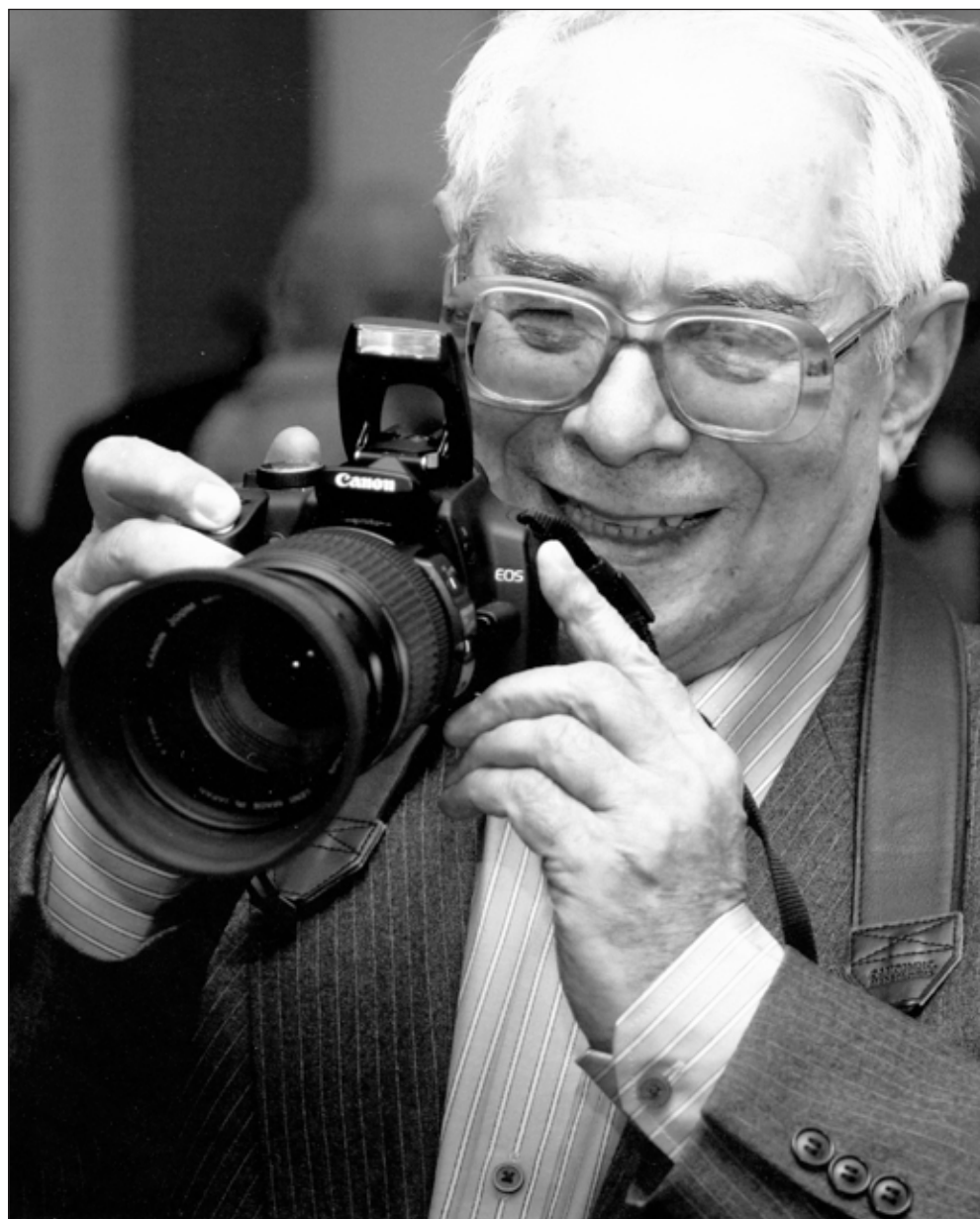


**Кетков
Юлий
Лазаревич.
Учёный.
Учитель.
Личность.**



Kenji

**Кетков Юлий Лазаревич.
Учёный. Учитель. Личность.**

Нижний Новгород
Растр-НН
2015

УДК 001
ББК 72.3 (2 Рос - 2 Ниж)
К-37

Составители:
И.В. Кеткова
Н.Г. Панкрашкина
А.Н. Рябов

Под редакцией **А.Ю. Кеткова**

Кетков Юлий Лазаревич. Учёный. Учитель. Личность / под ред. А.Ю. Кеткова – Нижний Новгород: Растр-НН, 2015 – 344 с.
ISBN

Книга посвящена памяти выдающегося нижегородского учёного и преподавателя, стоявшего у истоков разработки и программирования первых отечественных ЭВМ, доктора технических наук, заслуженного профессора Нижегородского университета, профессора кафедры математического обеспечения ЭВМ факультета Вычислительной математики и кибернетики Нижегородского государственного университета им. Н.И. Лобачевского Юлия Лазаревича Кеткова.

За почти 55-летний срок своей преподавательской деятельности, Ю.Л. Кетков обучил тысячи студентов программированию на языках Бейсик, Паскаль, Си, работе с разнообразным прикладным программным обеспечением.

В издание включены мемуары и воспоминания самого Юлия Лазаревича, его отдельные статьи, публикации и доклады, а также воспоминания о нём его родных, близких, коллег и учеников.

ISBN

УДК 001
ББК 72.3 (2 Рос - 2 Ниж)

**ЮЛИЙ ЛАЗАРЕВИЧ
КЕТКОВ**

Предисловие

Юлий Лазаревич Кетков – выдающийся нижегородский учёный и замечательный преподаватель. Нет ни одной области, связанной с программированием, развитие которой в Нижегородском регионе происходило бы без его участия, начиная от школьных олимпиад и заканчивая защитой диссертаций. Он не только учил студентов, занимался научно-исследовательской работой, был разносторонним и талантливым человеком, он был носителем истории отечественного программирования, и сам был живой историей этой области деятельности. В далёком 1956 г. среди шести студентов 5 курса физико-математического факультета Горьковского государственного университета им. Н.И. Лобачевского Юлий Лазаревич был направлен на стажировку в Московский государственный университет им. М.В. Ломоносова. Научным руководителем его дипломной работы стал А.А. Ляпунов, его наставниками – А.П. Ершов, Э.З. Любимский, М.Р. Шура-Бура, С.С. Камынин, В.С. Штаркман. Юлий Лазаревич Кетков не просто знал этих замечательных людей и учился у них, он успел рассказать об этой части истории становления отечественного программирования, эти рассказы составляют большую часть настоящей книги. Мемуары Юлия Лазаревича включают воспоминания о таких этапах становления российского программирования как разработка операционных систем первых ЭВМ, создание первых вузовских компьютеров, написание первых трансляторов высокоуровневых языков программирования, появление многопользовательских систем и многое другое.

Когда программирование появилось на свет, многие в среде математиков рассматривали его как прикладную математическую область, ставя математические модели во главу угла и в меньшей степени интересуясь аспектами, связанными с сопряжением программного обеспечения с оборудованием. Кетков же прекрасно знал аппаратные особенности ЭВМ и умело использовал эти знания при разработке соответствующего программного обеспечения. Попав после окончания университета по распределению в ГИФТИ, он участвовал в создании первой в стране вузовской цифровой вычислительной машины и первой ЭВМ, построенной в г. Горьком – машине ГИФТИ, где его интерес к аппаратному обеспечению позволил решить ряд вопросов, направленных на улучшение быстродействия. Юлий Лазаревич собрал коллекцию узлов и агрегатов вычислительных машин от первых систем до современных компьютеров, постоянно демонстрировал их студентам на лекциях, а впоследствии его коллекция стала основой экспозиции музея факультета вычислительной математики и кибернетики. Именно восприятие ЭВМ как симбиоза аппаратного и программного обеспечения позволяло ему

разрабатывать эффективные алгоритмы и писать программы, достойные подражания.

Юлий Лазаревич Кетков был последним из когорты разработчиков программного обеспечения для одного из самых знаменитых и титулованных отечественных компьютеров – БЭСМ-6, причём он не только принимал непосредственное участие в разработке ОС ИПМ, но и смог рассказать об этом в своих воспоминаниях.



Р.Г. Стронгин. Фото Ю.Л. Кеткова

Юлий Лазаревич стал инициатором и руководил разработкой первого в стране транслятора с языка Бейсик для ЭВМ типа М-20, что позволило поднять планку практического освоения ЭВМ на новую высоту. Как исследователь, он участвовал в решении сложнейших задач того времени, покрывавших область от космических полётов до эффективного раскроя корпусов судов при их проектировании. При его непосредственном участии в вычислительном центре ГГУ появился первый учебный терминал-класс в те времена, когда многопользовательские системы ещё только зарождались, и работал этот класс именно на Бейсике. Юлий Лазаревич любил вычислительное дело всем сердцем и

делился своей любовью – из-под его пера вышло множество книг, учебников и пособий, посвящённых как языкам программирования, особенно любимому Бейсику, так и сложным прикладным системам и свободно распространяемому программному обеспечению. Не одно поколение студентов училось, учится и будет учиться по его книгам.

Предлагаемая читателю книга содержит воспоминания не только коллег, но и учеников Юлия Лазаревича. За годы своей преподавательской деятельности он дал знания сотням специалистов, связавших свою жизнь с разработкой программного обеспечения, информационными технологиями и вычислительной техникой. Собранный материал предоставлен людьми, хорошо знавшими Юлиа Лазаревича, и рассказывает о его характере и личности не меньше, чем об успехах в научной деятельности. Его увлечения и хобби представлены не только в воспоминаниях коллег, учеников, родных и близких ему людей, но и в собрании фотографий, многие из которых сделаны самим Юлием Лазаревичем.

История Нижегородского государственного университета им. Н.И. Лобачевского – это история тех людей, которые внесли определяющий вклад в его развитие и становление. К этому ряду выдающихся деятелей науки и образования относится и Юлий Лазаревич: формирование нижегородской школы программирования было бы невозможным без его участия, и тысячи его студентов, многие из которых сейчас живут и работают по всему миру, демонстрируют высочайший профессионализм и превосходный уровень знаний.

Р.Г. Стронгин

Вице-президент Российского Союза ректоров,
Президент ННГУ,
заведующий кафедрой математического обеспечения ЭВМ,
доктор физ-мат. наук, профессор

Из когорты легендарных личностей

Первокурсникам факультета ВМК еще предстоит овладеть опытом программирования. Следует знать, что высокообразованный программист - это не только безукоризненный специалист в определенной области программирования, но и человек, имеющий широкий взгляд на современное программирование независимо от языков и платформ, которые он конкретно использует. Кроме того, он уверенно ориентируется в других сферах науки и искусства, знает своих корифеев, постоянно учится и обновляет свои знания, гибко мыслит, имеет абстрактное, системное и экспериментальное мышление.

Профессор Юлий Лазаревич Кетков относится к числу людей, встреча с которыми оставляет неизгладимое впечатление. Прежде всего следует отметить, что он входит в когорту легендарных личностей, которые творили историю отечественного программирования. Находясь еще на студенческой скамье, он был направлен в Москву в Институт прикладной математики АН СССР на стажировку, где принимал самое активное участие в разработке первых отечественных операционных систем. И это всегда впечатляло - одно дело читать про родоначальников, а Юлий Лазаревич был рядом, можно было пообщаться и посоветоваться.

Первое мое знакомство состоялось в середине 1970-х годов, когда нужно было выбирать кафедру для специализации. Я проходил обучение на многопользовательской системе «Студент», разработка которой была выполнена под руководством Юлия Лазаревича. Сейчас трудно себе представить, но вычислительные системы того времени выполняли только десятки тысяч операций в секунду (сейчас миллиарды) и имели крайне ограниченный объем памяти. Разработать на таких системах многопользовательскую работу, когда одновременно принимали участие 8-10 человек, можно было только при самом виртуозном владении секретами системного программирования. Оперативная работа в системе значительно повышала качество обучения. Важно также отметить, что эта система «Студент» являлась одной из первых подобного типа в стране и именно с этой системы началась эра терминалов-классов в Нижегородском университете.

Помню, как Юлий Лазаревич пригласил ряд моих сокурсников для выполнения масштабного проекта по разработке транслятора с алгоритмического языка Basic. Это был опять один из первых примеров, когда для выполнения курсовых и дипломных работ использовалась не какая-то учебная задача, а реальный профессиональный проект. И кроме того, это был первый пример командной разработки сложного программного обеспечения.

И здесь, конечно, надо отметить, что Юлий Лазаревич был фактически отечественным «властителем» языка Basic. Это относительно простой

язык программирования и, что является чрезвычайно важным, диалоговый язык, когда любая ошибка в операторе программы могла быть выделена транслятором, и можно было в оперативном режиме эти ошибки поправить. Разработка программы на языке Basic превращалась в увлекательнейшее взаимодействие с вычислительной машиной. Программирование становилось чрезвычайно интересным занятием.



В.П. Гергель и Ю.Л. Кетков

Юлий Лазаревич много сделал для использования языка Basic при обучении программированию. Разработаны были первые компиляторы, написаны несколько учебников. И это обучение прошли тысячи студентов Нижегородского университета. Пожалуй, без какого-либо преувеличения можно сказать, что если кто-то не знает языка Basic, тот не является программистом.

Здесь особо можно отметить издательскую деятельность Юлия Лазаревича. Им подготовлено и опубликовано более 10 монографий и учебников по самой широкой тематике от учебников по языку Basic до аналитических изданий по системам визуального программирования. А «Школьная энциклопедия по информатике», подготовленная совместно Ю. Кетковым и Д. Шапошниковым, стала просто бестселлером в стране, и не только для школьного образования, но и для университетского обучения программирования.

И, конечно, следует отметить, что Юлий Лазаревич являлся прекрасным преподавателем, пожалуй, он был один из самых любимых лекторов у студентов факультета вычислительной математики и кибернетики. Первоклассное впадение учебным материалом и дружественное отношение к студентам - особенно было важно, поскольку Юлий Лазаревич читал лекции для студентов первого курса. Он крайне редко ставил неудовлетворительные оценки, может быть, даже несколько завышал выставяемые баллы - жалел студентов, а может быть, помогал им поверить в свои силы.

В продолжение своей преподавательской деятельности Юлий Лазаревич в течение нескольких десятков последних лет занимался организацией олимпиад по программированию от факультетского уровня до уровня областных олимпиад для школьников и студентов. Здесь он охотно общался с талантливой молодежью, наиболее увлеченной программированием и информатикой. Конечно, не все побеждали в олимпиадах, но участие в них давало еще один импульс для повышения интереса к компьютерным технологиям. И, я думаю, многие из участников олимпиад связали свою будущую судьбу с нашей увлекательнейшей профессией - программированием во многом под влиянием Юлия Лазаревича.

И в завершение воспоминаний, конечно, хочется сказать о работе Юлия Лазаревича на кафедре математического обеспечения ЭВМ факультета ВМК. Юлий Лазаревич - один из сотрудников, которые во многом определяют облик кафедры как основного учебного подразделения Нижегородского университета по подготовке кадров по программированию и компьютерным технологиям. Его отличал высокий профессионализм, основанный на многолетней успешной деятельности по разработке сложных программных систем. Можно отметить его абсолютную преданность профессии. Вот и в свой последний день он, видно уже из последних сил, пришел на экзамен, хотя мы его и отговаривали. И еще - замечательные человеческие качества - доброту, внимание к коллегам,

готовность всегда прийти на помощь. И не было ни одного случая, чтобы Юлий Лазаревич на кого-то повысил голос: ни на коллег, ни на студентов (даже самых заядлых двоечников). С другой стороны, нельзя сказать, что мы всегда были с ним согласны. Бывало и спорили - скорее всего, не спорили, а устраивали оживленные дискуссии по тем или иным вопросам, что очень способствовало развитию творческой атмосферы на кафедре.

Это всего лишь несколько штрихов к характеристике нашего дорогого коллеги Юлия Лазаревича Кеткова. В памяти тысячи больших и маленьких моментов - но разве все это опишешь? Я надеюсь, что и мои коллеги осветят многие события, связанные с Юлием Лазаревичем. Но уже ясно главное - Юлий Лазаревич сыграл огромную роль в формировании в Нижегородском университете системы подготовки профессиональных программистов - дело, которому он отдал всю свою жизнь; и есть уверенность, что это наше наследие будет успешно развиваться и впредь.

В.П. Гергель,

Декан факультета ВМК ННГУ
директор НИИ ПМК
доктор техн. наук, профессор

Газета «Нижегородский университет»,
№ 08, сентябрь 2014 год

Краткая биография Юлия Лазаревича Кеткова

Кетков Ю.Л. – один из первых профессиональных программистов в городе Горьком, доктор технических наук, имеющий звание «Заслуженный профессор Нижегородского университета», профессор кафедры математического обеспечения ЭВМ факультета Вычислительной математики и кибернетики Нижегородского государственного университета им. Н.И. Лобачевского, заведующий отделом НИИ Прикладной математики и кибернетики родился 17 июля 1935 г. в городе Иркутске.



Его отец Лазарь Иванович Кетков, из украинских болгар, вынужден был с женой и старшим сыном Ремом покинуть Киев, где он в 30-е годы работал в аппарате ЦК ВЛКСМ Украины, и поселиться под Иркутском. Мать Ю.Л. Кеткова Голда Шмулевна Фрум закончила агрономический факультет Одесского университета, где на болгарском факультете учился её будущий муж. Незадолго до Великой Отечественной войны семья из-под Иркутска переехала в

болгарское село Андреевка Запорожской области. В первые дни войны отец ушёл на фронт, а семья оказалась в эвакуации за Уралом. Там, в поселке Кувандык Оренбургской области, Ю.Л. Кетков пошёл сразу во второй класс местной школы.

Перед окончанием войны мать с двумя сыновьями переехала в Горький к своей сестре Ф.Ш. Коренман, жене профессора И.М. Коренмана. Здесь оба брата закончили с медалями школу № 8. Старший Рем поступил в Физико-математический институт в Москве, младший – в 1952 г. на физико-математический факультет Горьковского университета, где уже с 1954 г. начинают складываться предпосылки для формирования новой специализации - вычислительной математики.

В начале 50-х годов Горьковский университет располагал мощным научным потенциалом в лице академика А.А. Андропова и его школы теории колебаний, предопределившей не случайность появления в Горьком первого в стране факультета вычислительной математики и кибернетики в 1963 г. Особая роль принадлежала профессору Ю.И. Неймарку, научное предвидение которого в сочетании с авторитетом учёного-механика придало импульс развитию процесса обучения новой дисциплине. Его курс лекций по методам вычислений имел продолжение в программе курсов по цифровой и аналоговой вычислительной технике. Всё это повышало уровень самой математической культуры.

Площадка фундамента значительно расширилась, когда из Москвы стали приглашать первых и немногих пока специалистов в области программирования. Студенты физмата имели возможность слушать лекции и участвовать в семинарах профессора МГУ А.А. Ляпунова, заложившего основы вычислительной математики в СССР, одного из создателей теорий математического обеспечения и программирования, заведующего отделом программирования в Отделении прикладной математики при Математическом институте им. В.А. Стеклова, а также молодых ученых и аспирантов МГУ, с которыми у Ю.Л. Кеткова позднее сложились товарищеские отношения. В 1956 г. руководство ГГУ приняло решение отправить группу студентов на стажировку в МГУ по специализации «вычислительная математика», в их числе был Ю.Л. Кетков. Там он и работал над своим дипломом под руководством А.А. Ляпунова. После успешной (закрытой) защиты начинающий программист возвращается в Горький и начинает трудиться в структуре ГИФТИ. С 1957 до 1964 г. он последовательно прошёл ступени карьерного роста, начиная от и.о. старшего научного сотрудника до главного инженера-конструктора Вычислительного центра.

В это время в ВЦ завершалась настройка небольшой цифровой ЭВМ с названием «машина ГИФТИ». Главной задачей молодых программистов было создание базового программного обеспечения. Ю.Л. Кеткову удалось придум-

мать универсальную схему, экономящую количество умножений, и данная схема прошла свою апробацию в новых стандартных программах.

Первая публикация, содержащая результаты этой работы, появилась в 1958 г. Когда в 1961 г. ВЦ приобрел одну из лучших ЭВМ первого поколения М-20, стала очевидной необходимость не только повышения квалификации программистов, но и расширения поиска применения новых возможностей. Вскоре к программистам из ВЦ обратились сотрудники НИИ Технологии машиностроения Ростова на Дону. Предложенная проблема сводилась к усовершенствованию автоматизации проектирования корпусов судов. За год удалось построить несколько версий универсальных интерпретирующих систем радиосографического метода. Работа открыла перспективу совершенствования способов задания криволинейных поверхностей и выдвинула ряд новых математических задач. В отечественной литературе это была первая работа в области сплайн-аппроксимации. Результаты данной работы послужили основой для диссертации на соискание учёной степени кандидата физико-математических наук, успешно защищённой в 1966 г.

С 1965 г. трудовая деятельность Ю.Л. Кеткова (сначала в качестве зав. лабораторией) оказалась надолго связана с вновь образованным НИИ Прикладной математики и кибернетики, где отдел № 4 возглавлял С.И. Альбер. Именно здесь начинается освоение новых систем программирования – трансляторов с алгоритмических языков, пакетов графических программ ГРАФОР, расширения интерпретирующей системы ИС-22 и разработка новых стандартных программ. Юлий Лазаревич стал инициатором и руководителем создания первой в стране диалоговой системы коллективного пользования на базе алгоритмического языка Бейсик. Он же написал первую в СССР монографию по Бейсику, составил толковый словарь данного языка, насчитывающий более 1000 операторов и функций.

Значительное событие в профессиональной судьбе Ю.Л. Кеткова произошло в 1967 г., когда он был привлечён к сотрудничеству с Математическим отделением Института математики им. В.А. Стеклова над проектом новой операционной системы ОС ИПМ БЭСМ-6 и где ему предстояло работать над наиболее важным компонентом системы – монитором. Почти полтора года он сотрудничал с выдающимися специалистами в области программирования: М.Р. Шурай-Бурой, Э.З. Любимским, С.С. Камыниным, В.С. Штаркманом, и другими, приобретая опыт участия в масштабном проекте государственного значения.

С 1969 г. он возглавляет отдел математического обеспечения ЭВМ в НИИ ПМК. Под его руководством отдел успешно работает над эксплуатацией Бейсик-компилятора и аналогичной версией, работающей под управлением MS-DOS. Подготовка программного обеспечения проектно-конструкторских за-

дач, стоящих перед крупнейшими в области предприятиями авиационной промышленности, становится приоритетной в планах отдела в 70-е годы. По сути это был тот вид инновационной деятельности, который обеспечивал эффективность проектов промышленного развития.

Новый круг проблем, определивший характер деятельности отдела и его руководителя, был обусловлен применением графопостроителей. Ю.Л. Кетков принимает участие в качестве заместителя научного руководителя профессора Ю.Г. Васина в работе по исследованию проблем автоматизированной обработки сложной графической информации о местности для обеспечения наземных и бортовых вычислительных комплексов. Под руководством Юлия Лазаревича созданы подсистемы вывода топографических и морских карт, а также системы создания библиотек описания дискретных знаков и символов картографических шрифтов. За эту работу коллективу разработчиков, в их числе и Ю.Л. Кеткову, была присуждена премия Совета министров СССР по кибернетике за 1990 г.

Ю.Л. Кетков возглавил работу по созданию программного обеспечения информацией в многомашинных комплексах технических средств АКС. В 1992 г. он защищает докторскую диссертацию на тему: «Создание инструментальных программных средств для разработки диалоговых систем САПР и АСНИ». Научная зрелость Ю.Л. Кеткова подтверждена обширным списком научных трудов – монографий, статей, выступлений на конференциях и съездах высокого уровня. Этот список публикаций включает 150 работ в открытой печати, в том числе 15 монографий, и 150 отчетов по научно-исследовательским и хозяйственным работам. Он был членом оргкомитета ряда Всесоюзных и Всероссийских научных конференций, входил в состав редколлегии двух межвузовских сборников, был членом двух советов по защите докторских диссертаций. В 1994-96 гг. его имя значилось в списке выдающихся учёных России, а в 1997 г. его избрали действительным членом Международной академии информации.

Педагогический стаж Ю.Л. Кеткова измерялся почти 55-летним сроком. С 1957 г. на правах совместителя он начинает читать первые лекции по программированию задолго до создания факультета ВМК. Традиционная преемственность научной школы А.А. Андропова генетически реализовалась сначала на кафедре «Теория управления и динамики машин», возглавляемой его учеником Ю.И. Неймарком, а с 1973 г. на кафедре «Математическое обеспечение ЭВМ» под руководством ученика Ю.И. Неймарка – Р.Г. Стронгина.

С этого года и до конца жизни Юлий Лазаревич являлся сотрудником данной кафедры, где формировались кадры школы отечественных программистов, признанной не только в нашей стране, но и за рубежом. Эта кафедра была по-

следним местом работы Ю.Л. Кеткова. Ещё в 1971 г. он был утвержден в звании доцента, а в 1994 г. занимает должность профессора.

Ю.Л. Кетков был неординарным педагогом. Он обладал даром логично и просто доводить до аудитории материал любой сложности и культивировал у своих учеников умение ясно мыслить и адекватно выражать свою мысль. Увлечённость предметом в соединении с высоким профессионализмом он щедро передавал аудитории, поэтому его лекции неизменно пользовались успехом и нередко студенты провожали своего преподавателя аплодисментами. Вероятно, не было такой научной проблемы в программировании, которая бы не нашла отражения в содержании его общего курса лекций и спецкурсов: алгоритмические языки, архитектура ЭВМ, ассемблеры, машинные коды, операционные системы, графические пакеты, редакционно-издательские системы, общий курс по программированию на Паскале, С и С++.

Много сил и душевного тепла он отдавал проведению олимпиад по информатике для школьников и студентов, особенно на том этапе, когда руководство олимпиадой предполагало совместную творческую работу жюри при составлении заданий. У Ю.Л. Кеткова были свои критерии при скрупулёзном рассмотрении каждой конкурсной работы. Для него большое значение имела и техническая готовность терминал-класса, он лично её контролировал перед началом конкурса. Кстати, под его руководством был создан первый в стране учебный терминал-класс, который на протяжении многих лет служил базой для подготовки студентов разных факультетов ННГУ в области информатики. Его руководство олимпиадами было отмечено Почётной грамотой Министерства образования Российской Федерации. Список его поощрений, благодарностей и наград пополнялся почти ежегодно. Среди них: «Почётный работник Нижегородского государственного университета», медаль «За доблестный труд», «День советской науки», знаки «Победитель соцсоревнования», «Ударник 10-й пятилетки», «За отличные успехи в работе» и другие. Почётное место в этом ряду занимал знак «За освоение целинных земель».

Юлий Лазаревич был человеком разносторонних интересов и способностей и всегда стремился к наибольшей их реализации. Занимаясь легкой атлетикой, он стал чемпионом города по прыжкам в высоту, затем пришло увлечение подводной охотой, фотографией, походами за грибами, собиранием коллекции музыкальных записей.

Юлий Лазаревич Кетков скончался 11 января 2014 года.

МЕМОУАРЫ

ИЗБРАННЫЕ СТАТЬИ

Пятьдесят лет с компьютером

Не знаю, для чего и для кого я пишу эти заметки. Может быть, это – попытка посмотреть на себя со стороны прожитых лет и как-то оценить свои деяния. Какое-то количество деревьев я посадил, двух сыновей вырастил (не один, конечно, на то и семья). Немного поучаствовал в строительстве дома (не своего), построил пару туалетов. А останется ли после меня какой-то след в науке кроме десятка быстро устаревающих книг? Или это сродни кругам на воде после брошенного камня? Не знаю. Время рассудит.

1. Последние годы учебы в ГГУ. Стрела-1

В сентябре 2006 года исполнилось ровно 50 лет с того дня, когда я впервые не только увидел настоящую ЭВМ, но и начал на практике осваивать свою будущую профессию программиста. Произошло это в результате договоренности между сотрудниками Горьковского государственного университета (ГГУ) и Московского государственного университета (МГУ). Шесть студентов 5-го курса физико-математического факультета – В.А. Бебихов, Т.Е. Бочкарева, И.А. Виткина, Ю.Л. Кетков, В.М. Корнилова и Ю.А. Первин, которые рискнули специализироваться по новой для ГГУ специальности «вычислительная математика», были направлены на годичную стажировку в Москву.

Этому предшествовали следующие события. Начиная с 1954 г. по приглашению со стороны ведущих ученых Горьковского исследовательского физико-технического института (ГИФТИ), физико-математического и радиофизического факультетов ГГУ в наш город начали приезжать московские ученые, закладывавшие основы новой науки – кибернетики. Годом ранее не без помощи академических ретроградов, запретивших на государственном уровне развитие генетики в СССР, кибернетика в нашей стране была объявлена «буржуазной лженаукой».

Самым известным среди гостей был профессор Алексей Андреевич Ляпунов. Он работал на кафедре Вычислительной математики на мехмате МГУ и по совместительству заведовал отделом программирования в Отделении При-

кладной Математики (ОПМ) при Математическом институте АН СССР имени акад. В.А. Стеклова. А.А. Ляпунов был достаточно известным исследователем в области теории множеств. Начав в новой для себя роли работу на одной из первых отечественных ЭВМ «Стрела-1», Алексей Андреевич сумел предугадать влияние вычислительной техники (ВТ) на развитие различных научных направлений. Он активно отстаивал позиции гонимых наук, и ему удалось сплотить многих ученых – генетиков, физиологов, лингвистов, математиков. Под его руководством с 1955 г. при кафедре Вычислительной математики на мехмате МГУ работал научный семинар по смежным вопросам кибернетики и физиологии. Часть семинаров проходили на квартире А.А. Ляпунова, и сопровождались они непременно чаепитием с пирожками, что дополнительно привлекало молодых слушателей. В 1955 г. в журнале «Природа» была опубликована первая статья А.А. Ляпунова об использовании вычислительных машин для перевода с одного языка на другой (в соавторстве с О.С. Кулагиной). Вскоре журнал «Вопросы философии» опубликовал его статью «Основные черты кибернетики» (в соавторстве с С.Л. Соболевым и А.И. Китовым).

Вместе с А.А. Ляпуновым к нам приезжали его коллеги из ОПМ, аспиранты и преподаватели МГУ. Среди них – С.В. Яблонский, заложивший основы многозначной логики, А.Г. Витушкин, занимавшийся исследованием сложности задачи табуляции функций, М.Л. Цетлин, разрабатывавший биопротез руки, О.С. Кулагина, трудившаяся над программами автоматического перевода с французского языка, и многие другие. А.А. Ляпунов познакомил нас с Игорем Андреевичем Полетаевым, автором первой отечественной книги по кибернетике «Сигнал», изданной с большим опозданием в 1958 г. Семинары, которые обычно проходили в 52-й аудитории старого здания ГГУ (ул. Свердлова, 37), привлекали не только преподавательский состав физмата и радиофака. Среди слушателей было довольно много студентов. На одном из семинаров А.А. Ляпунов рассказал о придуманных им операторных схемах программ – предвестниках будущих трансляторов. А вскоре он привез своего ученика Ю.И. Янова, который продолжил рассказ об операторных схемах и их преобразованиях. Иногда на этих семинарах появлялись и одиозные биологи, которые в лице профессора Е.М. Воронцова не менее яростно отстаивали признанные властями позиции академика Т.Д. Лысенко. Упоминание о генетике и кибернетике приводило их в бешенство.

Эти семинары оказали свое влияние на решение указанной шестерки студентов физмата специализироваться в области вычислительной математики. На четвертом курсе для знакомства с этим предметом вполне хватало допотопных арифмометров с ручным приводом (типа «Феликс») и более современных электромеханических калькуляторов немецкого производства («Reinmetall», «Mercedes»). Знакомству с этой техникой, с практическими методами решения

уравнений, с учетом распространения погрешности вычислений мы обязаны Юрию Исааковичу Неймарку. Но электронных вычислительных машин в то время не только в ГГУ, но и в городе Горьком еще не было. Поэтому и было принято решение отправить первую группу «вычислителей» в Москву. Благо расходы со стороны ГГУ были минимальными – проезд и суточные нам не оплачивались. Основная материальная помощь заключалась в переводе студенческой стипендии на почтовое отделение МГУ. Правда, руководство мехмата позаботилось о предоставлении нам мест в студенческом общежитии в Черемушках. Лекции за 5-й курс мы посещали вместе со студентами мехмата МГУ, а производственная и дипломная практика проходила в разных организациях – ВЦ АН СССР, ВЦ МГУ, лаборатории управляющих машин и ОПМ МИ АН СССР. Мы с Первиным попали в ОПМ и проходили практику в 9-ом отделе. А.А. Ляпунов был официальным руководителем наших дипломных работ.

Тема дипломной работы Ю.А. Первина была открытой – ему предстояло написать первую в СССР игровую программу «Морской козел» (разновидность игры в домино). В те времена сотрудники ОПМ в обеденный перерыв сражались не только в шахматы и пинг-понг, но и «забывали козла». А моя дипломная работа оказалась закрытой. Мне поручили написать программу для решения обратной задачи Штурма-Лиувилля. Однако исходные данные для нее были связаны с секретной тематикой (в те времена ОПМ был головной организацией по разработке методов и программ для решения задач ядерной физики). По ходу выполнения дипломной работы мне пришлось познакомиться с численными методами решения систем обыкновенных дифференциальных уравнений и запрограммировать в машинных кодах Стрелы-1 метод Рунге-Кутты. Но гораздо более интересной оказалась возможность воспользоваться первыми средствами автоматизации программирования, разработанными в 9-ом отделе – программирующей программой ПП-2 [1] и программой автоматического присвоения адресов ПАПА [2]. Эти средства были разработаны в 1955-56 гг. и находились еще в стадии начального и не очень активного освоения программистами других отделов. Поэтому появление в 9-ом отделе своего подопытного кролика было встречено очень доброжелательно. ПАПА позволяла автоматизировать процесс замены условных (символических) адресов в машинных командах на автоматически вычисляемые физические адреса. В определенном смысле, текст программы походил на строки автокодов (ассемблеров) за тем исключением, что в машинных командах указывались числовые коды операций. ПП-2 позволяла автоматизировать и сам процесс написания программы по соответствующей операторной схеме. По сути дела, она явилась прообразом трансляторов с алгоритмических языков, появившихся в нашей стране спустя 5 лет.

Во время пребывания в Москве я познакомился с некоторыми однокурсниками моего старшего брата. Большинство из них тогда учились в аспирантуре, совмещая учебу с проведением учебных занятий и работой. Именно они заложили базу профессионального программирования в нашей стране.

Среди них аспирант А.А. Ляпунова – Андрей Петрович Ершов, который уже читал лекции по программированию для студентов мехмата. Впоследствии А.П. Ершов превзошел своего учителя и стал академиком, возглавившим клан советских программистов. Под его руководством в ВЦ АН СССР была создана одна из первых программирующих программ для ЭВМ типа БЭСМ, разработаны расширения алгоритмического языка АЛГОЛ-60 и созданы соответствующие АЛЬФА-трансляторы для БЭСМ-6. Он очень много сделал для становления информатики как предмета обучения в школе. До конца дней своих А.П. Ершов возглавлял отдел программирования в Вычислительном центре Сибирского отделения АН СССР.

Анатолий Георгиевич Витушкин, аспирант А.Н. Колмогорова, до поступления в МГУ был выпускником суворовского училища. На одном из занятий у него в руках взорвался запал боевой гранаты. В результате будущий офицер лишился зрения и двух пальцев на левой руке. А советская математика в его лице приобрела блестящего ученого – одного из самых молодых членкорреспондентов АН СССР. Будучи слепым, А.Г. Витушкин неплохо играл в шахматы, на слух изучал и исполнял классические музыкальные произведения – фортепьянные концерты Грига, Рахманинова, Чайковского. Одним из его хобби была разработка читающего автомата для слепых. На входе автомата предполагалось установить миниатюрную телевизионную камеру, которая считывала страницу текста. Оптическая система должна была выделять в поле зрения отдельную букву, в образе которой логическим дешифратором распознавались такие графические элементы как горизонтальный, вертикальный или наклонный отрезок прямой, дуга окружности с той или иной ориентацией (выпуклость влево/вправо, вверх/вниз). Совокупность графических признаков выделенной буквы сравнивалась с эталонными образцами, и полученный таким образом двоичный код управлял шестеркой соленоидов со штырями-сердечниками, формировавших на выходе тактильный узор опознанного символа в системе Брайля. К разработке проекта этого автомата А.Г. Витушкин привлек и меня. Какое-то время меня даже зачислили в штат ОПМ в качестве лаборанта по совместительству.

Большую роль в моем профессиональном становлении сыграл Эдуард Зиновьевич Любимский, который в то время работал научным сотрудником 9-го отдела и назывался просто Сашей. Он был моим непосредственным опекуном как в освоении программирования на ЭВМ «Стрела-1», так и в использовании средств автоматизации программирования. Именно он вместе с Сергеем

Сергеевичем Камыниным и Михаилом Романовичем Шурой-Бурой разработал программирующие программы ПП-1 и ПП-2 для ЭВМ типа Стрела. Спустя несколько лет Э.З. Любимский возглавил коллектив, разработавший один из первых отечественных трансляторов ТА-2 с языка АЛГОЛ-60 для ЭВМ тип М-20. Именно ему вместе с С.С. Камыниным принадлежит идея создания алгоритмического машинно-ориентированного языка АЛМО и серии трансляторов, использовавших АЛМО в качестве языка-посредника. В 1966 г. он был одним из оппонентов на защите моей кандидатской диссертации. А через пару лет он пригласил меня в сборную команду программистов СССР, которая в 1968-69 гг. разработала операционную систему ОС ИПМ для БЭСМ-6. Многие идеи этой системы были положены в основу докторской диссертации Э.З. Любимского, который до настоящего времени является ведущим профессором кафедры математического обеспечения МГУ и одновременно заведует отделом в Институте Прикладной Математики РАН им. акад. М.В. Келдыша. (Э.З. Любимский умер в 2008 г. - *примеч. ред.*)

Многое я почерпнул из общения с другим научным сотрудником 9-го отдела Всеволодом Серафимовичем Штаркманом. Этому человеку были подвластны не только нюансы программирования в машинных кодах, но и детали инженерной реализации узлов и блоков ЭВМ. Он принимал активное участие в проектировании системы команд на первых отечественных ЭВМ типов Стрела и М-20. Его перу принадлежит одна из моих первых книг по программированию «Система команд быстродействующей электронной вычислительной машины Стрела-1». Он был руководителем группы, разработавшей отечественный транслятор языка машинных команд для ЭВМ БЭСМ-6. Автокод БЕМШ (Бочкова – Езерова – Морозова – Штаркман) был одним из главных средств разработки различных программных систем для флагмана нашей вычислительной техники. В руках В.С. Штаркмана я увидел первые фотографии обратной стороны Луны, которые были обработаны на БЭСМ-6 и только спустя несколько дней появились на страницах газет.

В 1956 г. нас очень удивил приказ по ОПМ, в котором сообщалось о награждении орденами группы ученых ОПМ, среди которых находились и сотрудники 9-го отдела. В их число входили и такие корифеи как зам. зав. отделом д.ф.-м.н., профессор М.Р. Шура-Бура, и почти наши ровесники Э.З. Любимский и В.С. Штаркман, которым тогда было по 24-25 лет. Позже мы узнали, что эти правительственные награды наши знакомые получили за успешное завершение одного из этапов ядерной программы СССР.

В то время штатная численность сотрудников ОПМ примерно в 7-8 раз превышала количество сотрудников МИ АН СССР, к которому было приписано отделение. Инженерно-технические подразделения, возглавляемые А.Н. Мямлиным, насчитывали до 300 сотрудников, обслуживавших первую серийную

ЭВМ «Стрела-1». Они обеспечивали круглосуточную работу перфораторных, холодильных и энергетических установок, профилактику, ремонт и эксплуатацию ЭВМ и ее периферийных устройств. Все это хозяйство располагалось в отдельном двухэтажном корпусе, находившемся под бдительным оком вооруженной охраны, подчинявшейся генералу КГБ. Ничего удивительного в этом нет. ОПМ был ведущим разработчиком математического и программного обеспечения систем, использовавшихся в ядерных центрах Сарова и Челябинска. Здесь же создавались программы управления полетами будущих космических аппаратов. Руководителями научных отделов ОПМ были академики Я.Б. Зельдович, А.Н. Тихонов, А.А. Самарский, член-корреспондент И.М. Гельфанд, доктора наук А.А. Ляпунов, Д.Е. Охоцимский, М.Р. Шура-Бура. А возглавлял отделение академик Мстислав Всеволодович Келдыш, избранный впоследствии Президентом АН СССР.

Таким образом, мы в нужное время оказались в нужном месте.

2. Машина ГИФТИ

2.1. Первая тройка программистов в г. Горьком

После окончания ГГУ пять из шести вычислителей были распределены на работу в ГИФТИ. Инна Виткина получила приглашение от А.П. Ершова и уехала на работу в только что созданный Вычислительный центр СО АН СССР. Я, Ю. Первин и В. Максимова были приписаны к инженерной группе, занимавшейся разработкой первой в Горьком ЭВМ, получившей название «Машины ГИФТИ». Т.Е. Бочкарева и В.А. Бебихов попали в группу, занимавшуюся эксплуатацией серийной аналоговой вычислительной машины МН-8, которую институт приобрел для исследования сложных динамических систем. Спонсором, обеспечившим покупку этого дорогостоящего оборудования, было Особое Конструкторское Бюро Машиностроения (ОКБМ), возглавляемое И.И. Африкантовым.

Проект первой в СССР вузовской цифровой вычислительной машины был задуман в 1955-56 гг. сотрудниками кафедры теории колебаний радиофизического факультета ГГУ и активно поддержан ее новым заведующим Николаем Александровичем Железцовым. Задолго до этого интерес к работам в новом направлении был проявлен его предшественником – академиком Александром Александровичем Андроновым. Ему довелось участвовать в работе одной серьезной комиссии, проверявшей целесообразность расходования довольно больших средств в лаборатории управляющих машин. Выводы комиссия сделала правильные, и впоследствии на базе лаборатории, возглавляемой чл.-корр.

АН СССР И.С. Бруком и разработанной там ЭВМ М-2, был создан Институт электронных управляющих машин (ИНЭУМ).

Первый период был более характерен работами теоретического направления. К ним относятся блестящая дипломная работа Марка Исааковича Фейгина, связанная с исследованием динамики поведения триггера (1952 г.), проект арифметического устройства ЭВМ последовательного действия (Михаил Яковлевич Эйнгорин, 1954 г.), система команд и архитектура ЭВМ с двухуровневой памятью (Аркадий Моисеевич Гильман, 1955 г.). Две последние работы были представлены на Всесоюзной конференции «Пути развития советского математического машиностроения и приборостроения», состоявшейся в 1956 г.

Воплощение двоичной логики в железо началось с цифровой лабораторной работы – специализированной машины со странным названием «7×7». Автором этого проекта был М. Фейгин, который довел свой замысел до ума в 1953 г. вместе с бригадой студентов младшего курса в составе И. Клибановой, Е. Сабаяева и А. Сергиевского. Машина «7×7» умела вычислять произведение трехразрядных двоичных чисел, и с ее помощью ассистент В.А. Дозоров наводил страх на студентов физмата и радиофака. Он подменял исправные элементы схемы на неисправные, а задача обучающихся состояла в локализации ошибочного диода по таблице получающихся результатов.

Наряду с учебными макетами подобного рода и глубокими теоретическими изысканиями на кафедре началась и более кропотливая экспериментальная работа по созданию отдельных узлов и блоков цифровой техники. В 1954-55 гг. довольно много дипломных работ (С. Буторин, А. Гончаров, Б. Караулов, Б. Кожинская и др.) было посвящено решению этих практических задач. Исторически сложилось так, что выпускники кафедры, посвятившие себя новой тематике, группировались вокруг Аркадия Степановича Тарановича (выпускника 1953 г.). В составе группы инженеров-разработчиков, включенной в штат ГИФТИ и активно поддерживаемой ее директором Яковом Никитичем Николаевым, появились Александр Михайлович Гончаров (выпускник 1955 г.), Марк Давыдович Брейдо, Нина Всеволодовна Жеглова, Григорий Дмитриевич Зарницын и Рафик Хасьянович Садеков (выпускники 1956 г.). Основной объем работ по изготовлению блоков машины ГИФТИ выпал на группу, опекаемую Зоей Семеновной Кечиевой. В ее составе работали техники и монтажники Леша Алексеев, Саша Аралов, Валя Блинничева, Лева Маркин, Вена Монахов, Толя Рожков и чертежница Дина Мануилова. В разработке силовых компонент (блоки питания, сетевые фильтры) принимал активное участие Виктор Иванович Королев. Дизайн пульта ЭВМ и разработка электронной схемы управления встроенным растровым дисплеем выполнены Геннадием Григорьевичем Денисовым.



Слева направо: Г.Д. Зарницын, А. Аралов, Р.Х. Садеков, Н.А. Железцов, Н.В. Жеглова, А.М. Гильман, М.Я. Эйнгорин, А.М. Гончаров, В.А. Дозоров

В 1957 г. общее руководство работами по созданию, монтажу и вводу машины ГИФТИ в эксплуатацию было поручено к.ф.-м.н. Артемию Сергеевичу Алексееву, который возглавил образованный к концу года Вычислительный центр ГИФТИ и руководил им, практически, до конца своей жизни. На фото выше представлены сотрудники ГИФТИ, так или иначе причастные к разработке проекта машины ГИФТИ.

В основу машины ГИФТИ был положен проект А.М. Гильмана, однако в процессе его реализации многие функциональные узлы подверглись серьезным изменениям. Машина ГИФТИ представляла собой универсальную ЦВМ последовательного действия с оперативной памятью из 2016 слов длиной по 32 бита. Специально для нее в ОКБМ был изготовлен магнитный барабан, вращавшийся со скоростью 6000 об/мин. На этом барабане помимо ячеек оперативной памяти были реализованы сверхбыстрые рециркуляционные регистры, позволившие довести скорость работы арифметического устройства до 6000 сложений в сек.



«Память» машины ГИФТИ – А.М. Гончаров и А.С. Тарантович

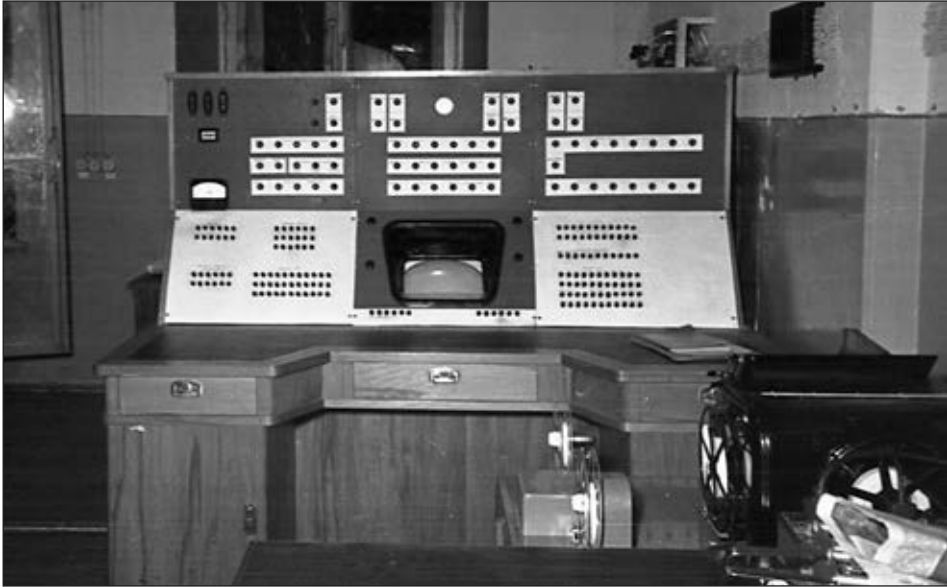
Замечу, что ранее разработанные отечественные ЭВМ обладали быстроедействием 100 оп/с (Урал-1), 2000 оп/с (Стрела-1) и 7000 оп/с (БЭСМ). Причем в двух последних компьютерах была реализована более дорогостоящая параллельная арифметика. В арифметическом устройстве машины ГИФТИ была реализована схема ускоренного умножения и смоделирована оригинальная схема ускоренного деления двоичных чисел. К разработке последней приложил руку и я [3]. Моя тяга к изучению «железа» ЭВМ, наверное, была инициирована общением с В.С. Штаркманом. Упомяну в связи с этим еще одну публикацию, направленную на оптимизацию конструкции дешифратора памяти [4].



М.Д. Брейдо и Г.Д. Зарницын за отладкой арифметического устройства

Общая производительность машины ГИФТИ сдерживалась медленной оперативной памятью. Однако система ее команд предусматривала довольно много операций типа регистр-регистр или память-регистр. Впоследствии такой подход стал основным в архитектуре машин третьего поколения – ЕС ЭВМ и СМ ЭВМ. Одной из интересных особенностей логики выполнения команд машины ГИФТИ была система автоматической модификации исполнительного адреса и управления приращением в индексном регистре. На пульте машины находилась электронно-лучевая трубка, предназначенная для просмотра содержимого регистров и ячеек оперативной памяти, используемая как своеобразный растровый дисплей. Ввод данных и программ производился с перфоленты со скоростью 300 знаков/с, тогда как электромеханический трансмиттер фото-телеграфного аппарата выжимал всего 7 знаков/с. Для вывода результатов вычислений использовался обычный рулонный телетайп РТА-51.

По сравнению с ЭВМ типа Стрела, занимавшей тогда машинный зал площадью в 300 кв.м. и потреблявшей более 300 кВт электроэнергии, машина ГИФТИ поражала своими скромными параметрами (комната в 25 кв.м., 560 ламп, потребляемая мощность – порядка 11 кВт).



Пульт машины ГИФТИ. Справа – телетайп РТА-50.



Одна из трех стоек машины ГИФТИ. Справа – типовые блоки.

Машина ГИФТИ оказалась пятой в стране универсальной цифровой вычислительной машиной вслед за ЭВМ БЭСМ (разработчик – ИТМ и ВТ, руководитель акад. С.А. Лебедев, кстати, нижегородец по происхождению), Стрела-1 (разработчик – Московский завод САМ, руководитель Ю.Я. Базилевский), М-2 (разработчик – Лаборатория управляющих машин АН СССР, руководитель чл.-корр. И.С. Брук), Урал-1 (разработчик – Пензенский завод САМ, руководитель Б.И. Рамеев). И она была первым компьютером, разработанным вузовскими учеными.

Первый коллектив пользователей машины ГИФТИ состоял из трех выпускников физмата – Ю.Л. Кеткова, В.М. Корниловой и Ю.А. Первина. Мы помогали инженерам доводить конструкцию ЭВМ, составляли первые тесты, занимались разработкой программ нулевого цикла – подпрограмм ввода/вывода числовой информации, вычисления элементарных функций, реализации численных методов интегрирования, решения задач линейной алгебры, систем обыкновенных дифференциальных уравнений. К новому 1958 г. на экране нашего дисплея появилась первая цифровая мультипликация: на фоне елочки, контуры которых были образованы неподвижными битовыми комбинациями, опускались снежинки – перемещающиеся ярко светящиеся точки (двоичные «единицы»). Электронная докладная, сопровождающая этот графический шедевр и адресованная начальнику Вычислительного центра, содержала требование повысить зарплату (зарплата программистов, зачисленных на должности и.о. старших научных сотрудников, составляла 880 руб. и не так уж и намного превосходила их студенческую стипендию).

Первые инженерно-технические задачи, которые решались на машине ГИФТИ, были связаны с исследованиями систем обыкновенных дифференциальных уравнений высокого порядка. Главным поставщиком задач такого рода была лаборатория, возглавляемая Н.А. Железцовым, которая по заданию ОКБМ разрабатывала и исследовала схемы управления ядерными реакторами. В силу закрытости этих работ лаборатория именовалась как «п/я 88». Поначалу такие задачи решались на большой аналоговой вычислительной машине МН-8, многочисленные блоки которой соединялись в соответствии с математической моделью исследуемого объекта. Коммутация блоков осуществлялась вручную с помощью проводников на специальной панели, занимала много времени, требовала тщательной проверки и настройки параметров интегрирующих усилителей. Затраты на операции такого рода занимали несколько дней, и до тех пор, пока исследовалось поведение одной системы, коммутационную панель МН-8 нельзя было занимать для набора другой схемы. Погрешности аналоговых блоков и нестабильность их параметров не всегда гарантировали требуемую точность решения.

На машине ГИФТИ ввод программы решения аналогичной задачи занимал считанные секунды и после получения многометровых распечаток с таблицами исследуемых функций пользователь мог неспешно их анализировать, освобождая компьютер для решения других задач. Первым моим клиентом был сотрудник «секретного ящика» Андрей Владимирович Сергиевский. Впоследствии он стал директором ГИФТИ и несколько лет проработал директором НИИ ПМК.

Следующую группу исследователей, вкусивших прелести численных методов решения дифференциальных уравнений, составили сотрудники Людмилы Николаевны Беллустинной, занимавшиеся качественным исследованием динамических систем. Вскоре появились и первые внешние заказчики – проф. С.А. Жевакин (НИРФИ), проф. И.И. Трянин (ГИИВТ), молодые аспиранты проф. А.Г. Угодчикова (ГИСИ) и др. К 1960 г. группа программистов в ВЦ ГИФТИ насчитывала в своих рядах уже более 20 человек, для которых я был неформальным начальником.

До 1961 г. машина ГИФТИ была в Вычислительном центре единственной цифровой ЭВМ, на которой успешно решались многие научно-технические задачи и воспитывались первые кадры горьковских программистов.

2.2. Как я бодался с Горнером

Для вычисления значений элементарных и специальных функций на каждой ЭВМ разрабатывается библиотека соответствующих стандартных подпрограмм. В большинстве случаев используются подходящие полиномы, с той или иной степенью точности аппроксимирующие нужную функцию:

$$y = f(x) \approx P_n(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_{n-2} x^2 + a_{n-1} x + a_n$$

Поэтому задаче вычисления значения полинома в заданной точке всегда уделялось большое внимание. Исторически сложилось так, что для ее решения все использовали классическую схему Горнера:

$$P_n = (\dots ((a_0 x + a_1) x + a_2) x + \dots + a_{n-1}) x + a_n$$

Эта схема требует в общем случае выполнения $(n+1)$ операции умножения и n операций сложения. Интуитивно чувствуется, что по числу операций схема Горнера является самой оптимальной, однако этот факт никогда ранее не доказывался.

В 1957 году была закончена разработка машины ГИФТИ, и мне предстояло написать для нее программы вычисления элементарных функций. Среди литературы, посвященной этим вопросам, мне попала на глаза свежая статья Джона Тодта «Мотивы для работы в области численного анализа», опубликованная в журнале «Математическое просвещение» (№ 75, вып. 1, 1957). В этой статье описывалась идея нового способа вычисления полинома 6-ой степени.

Речь шла о том, чтобы на предварительном этапе пересчитать коэффициенты a_1 исходного полинома в некоторые другие коэффициенты b_j , а затем воспользоваться следующей схемой:

$$\begin{aligned} \alpha_1 &= (x + b_1) \times x \\ \alpha_2 &= (\alpha_1 + x + b_2) \times (\alpha_1 + b_3) \\ \alpha_3 &= (\alpha_2 + b_4) \times (\alpha_1 + b_5) + b_6 \end{aligned} \quad (1)$$

Из условия $\alpha_3 = P_6 = x^6 + a_1 x^5 + a_2 x^4 + a_3 x^3 + a_4 x^2 + a_5 x + a_6$ и приравнивания выражений при одинаковых степенях x составлялись уравнения для нахождения неизвестных коэффициентов b_j . Среди полученных уравнений были и квадратные, которые могли не иметь вещественных корней. Но в тех случаях, когда по коэффициентам a_1 можно было вычислить вещественные b_j , схема (1) позволяла снизить количество операций на заключительной стадии подсчета значения полинома: вместо 6 умножений и 6 сложений (как это было в схеме Горнера) требовалось выполнить 3 умножения и 7 сложений. Предварительная обработка коэффициентов исходного полинома выполнялась один раз, зато при последующих многократных вычислениях по схеме (1) удавалось сократить количество наиболее медленных операций умножения.

Машина ГИФТИ, для которой мне пришлось создавать библиотеку стандартных подпрограмм, была вычислительной машиной последовательного действия, и операция умножения на ней выполнялась в 32 раза медленнее, чем операция сложения. Поэтому идея Тодта показалась мне очень привлекательной, и основное, что я в ней разглядел, – возможность повысить степень промежуточного полинома при очередном умножении не на 1 (как это происходит в схеме Горнера), а на 2 за счет умножения на квадратный множитель. Довольно быстро мне удалось построить похожие схемы для полиномов четной степени $n = 2k \geq 6$, в которых предварительная обработка всегда приводила к вычислению вещественных коэффициентов b_j . В один из приездов А.Г. Витушкина в Горький я рассказал ему о новых схемах, требовавших после предварительной обработки выполнения $[(n+1)/2] + [n/4]$ умножений и $(n+1)$ сложений. Вернувшись в Москву, он показал мои схемы своим будущим дипломникам Э.Г. Белаге и В.Я. Пану и предложил им тоже заняться этой задачей.

В результате этого «соревнования» Э.Г. Белаге удалось доказать, что применение предварительной обработки не позволяет построить схему вычисления полинома, использующую меньше, чем $[(n+1)/2] + 1$ умножение и n сложений. В общем, этот результат интуитивно предсказывал и я, т.к. за одно умножение к каждому операнду можно было добавить только по одному коэффициенту b_j , а их количество в общем случае (все a_1 линейно независимы) должно было точно равняться $n+1$. Для полиномов с комплексными коэффициентами Э.Г. Белага построил схему, требовавшую минимального количества операции. Однако для практических целей эта схема не годилась, т.к. операции ум-

ножения комплексных чисел существенно замедляли вычисления, да и каждое сложение удваивало количество тактов ЭВМ. Зато В.Я. Пан сумел построить схему предварительной обработки полинома с вещественными коэффициентами, которая позволяла найти вещественные b_j и требовала на заключительном этапе $[n/2]+2$ умножений и $(n+1)$ сложений. Моя схема и схемы Пана достаточно подробно описаны в книге из серии «Справочная математическая библиотека» – Математический анализ: Вычисление элементарных функций. – М.: Гос. изд. физ.-мат. литературы, 1963. За несколько лет до выхода этого справочника «соперники» постарались застолбить свои приоритеты [5-9]. Жаль только, что ни Пан, ни Белага не сослались на предоставленные им схемы, которые я придумал чуть раньше. Хотя после первых публикаций мы с Витей Паном переписывались и обменивались очередными достижениями в деле улучшения схемы Горнера. Последующие работы В.Я. Пана в этой области [10-13] составили основу его кандидатской диссертации.

2.3. Как хитрая запятая доплыла до IBM

Арифметическое устройство машины ГИФТИ было предназначено для выполнения операций над двоичными числами с фиксированной запятой. Система команд предусматривала три положения запятой:

- в начале числа, при этом максимальное по модулю число было равно $2-2^{-27}$;
- в середине числа, при этом максимальное по модулю число было равно $2^{14}-2^{-15}$;
- в конце числа, т.е. все числа были целыми.

Любое из них мало подходило для решения задач, в которых наблюдался большой разброс в диапазоне исходных данных и промежуточных результатов. Поэтому приходилось вводить различные масштабные коэффициенты. Полученные таким образом результаты решения задачи приходилось также вручную подвергать обратным преобразованиям. Для освобождения программистов от этой довольно нудной работы З.С. Баталовой, входившей в состав нашей группы, было предложено разработать подпрограммы, моделирующие выполнение операций с плавающей запятой. При этом в качестве формата исходных данных было выбрано стандартное для конструкций более мощных ЭВМ того времени двоичное представление:

$$x = -1^s \times m \times 2^p$$

Здесь

- s – знак числа, (0 для положительных чисел, 1 для отрицательных)
- p – двоичный порядок числа,

m – двоичная мантисса числа, обычно, нормализованная, т.е. $0.5 \leq m < 1$.

Мантиссы и порядок располагались в смежных ячейках памяти. Для мантиссы был выбран формат числа с фиксированной запятой, которое по модулю принадлежало диапазону $[2 \cdot 2^{-27}, 2^{-27}]$. Порядок представлялся как целое число, по модулю принадлежащее диапазону $[2^{29}-1, 0]$. Это позволяло существенно расширить диапазон обрабатываемых числовых данных и забыть о проблеме масштабирования переменных в программах. Зина Баталова успешно справилась с этой работой и ее первая публикация по информатике появилась в журнале «Известия вузов. Радиофизика» [14].

Так был найден разумный компромисс, позволивший отказаться от неприятной ручной работы ценой снижения общей скорости работы программы – моделирование операций с плавающей запятой требовало изрядного количества дополнительных машинных тактов. Подпрограммы с плавающей запятой облегчали процедуру решения многих прикладных задач. Однако накладные расходы на подключение соответствующих подпрограмм, особенно на ввод и вывод данных были достаточно большими. Кроме того, подпрограммы использовались для своих нужд изрядное количество быстрых регистров.

Проанализировав плюсы и минусы нового режима решения задач, я решил улучшить эксплуатационные характеристики подпрограмм для выполнения операций с плавающей запятой за счет:

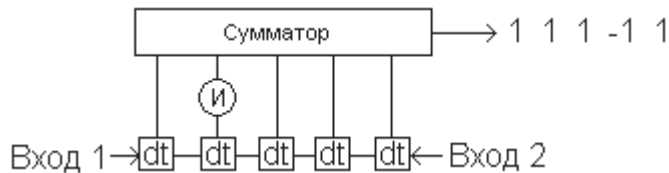
- сокращения объема подпрограмм;
- использования в них минимального количества ячеек специального запоминающего устройства (эти ячейки, выполненные на регистрах рециркуляции, обладали максимальным быстродействием, но их было очень мало);
- упрощения работы с данными на вводе и выводе.

В реализации этого плана мне помогла идея отказа от общепринятого формата представления числовых данных с плавающей запятой. После замены двоичного порядка p на десятичный порядок d ($x = -1^s \cdot m \cdot 10^d$), удалось упростить преобразование данных на вводе и выводе и еще больше расширить диапазон представимых данных. Мантисса исходных данных вводилась по стандартной схеме ввода чисел с запятой в начале числа, а порядок вводился в восьмеричной системе, т.е. не требовал специального преобразования. Так же просто был организован и вывод результатов вычислений. Таким образом, отпала необходимость в специальных подпрограммах ввода и вывода числовых данных. При использовании нового способа представления чисел объем исходной программы увеличивался в 2-4 раза, одновременно в 10-12 раз возрастало время решения задачи. Информация об этом подходе была опубликована в следующем номере журнала «Радиофизика» [15]. А спустя несколько

лет фирма IBM использовала аналогичную идею, заменив двоичный порядок шестнадцатеричным в своих знаменитых моделях IBM/360/370.

2.4. Коды Баркера

Одна из задач, которую мы пытались решить с помощью машины ГИФТИ, была связана с проектом новой радиолокационной станции, разрабатываемым на одном из предприятий г. Горького. Идея ее работы была основана на использовании кодов Баркера – дискретных кодов, состоящих из последовательности прямоугольных импульсов с синусоидальным заполнением в фазе и противофазе. Для упрощения задачи код Баркера можно представить себе как последовательность положительных и отрицательных импульсов. Например, при длине кода 5 такая последовательность задается цепочкой $1, 1, 1, -1, 1$. Сгенерировать такой сигнал можно с помощью следующей логической схемы:



Здесь используются пять задержек на некоторый фиксированный интервал времени dt , задающие требуемую частоту импульсов, инвертор «И», меняющий знак сигнала на своем входе, и сумматор. Если на Вход 1 подать импульс, то на выходе сумматора образуется нужная последовательность из пятерки положительных и отрицательных единиц. Сигнал, образованный на выходе сумматора, отправляется в сторону предполагаемой цели, отражается от нее и возвращается уже на Вход 2 этой же схемы. По результатам суммирования импульсов отраженного сигнала на выходе сумматора образуется следующая цифровая последовательность: $1, 0, 1, 0, 5, 0, 1, 0, 1$. Обратите внимание на тот факт, что центральный суммарный импульс в 5 раз превышает амплитуду остальных импульсов, отраженных от цели. По сути дела, это – функция автокорреляции радиолокационного сигнала. Однако отраженный сигнал возвращается на приемник локатора с неизбежными шумами, и его амплитуда может оказаться ниже уровня шума. Известно, что n -кратное суммирование белого шума увеличивает его амплитуду в \sqrt{n} раз. Поэтому при использовании кода Баркера с достаточно большим n сигнал от реальной цели можно было бы четко выделить на фоне шумовых помех. Дело было за малым – заказчика интересовал код Баркера для $n=30$.

Первая идея, которая пришла к нам в голову – написать программу, которая бы перебирала все комбинации битов 30-разрядного двоичного кода и строила

бы для них результирующий ответный сигнал. Но уже на сравнительно малых n мы убедились в том, что скорости машины ГИФТИ для перебора такого объема было явно недостаточно. Некоторые коды Баркера для $n = 7, 9, 11$ и 13 мы таким способом нашли, но мощности нашей ЭВМ на этом были исчерпаны. Почти параллельно с нами над этой проблемой трудились сотрудники Радиотехнического Института АН СССР Ю.В. Поляк и Р.В. Мошетов. Им удалось аналитически доказать, что для $n=4k\pm 1$ при $n>13$ кодов Баркера не существует [16]. Поэтому мы решили использовать аналогичный подход для кодов четной длины. Нам удалось доказать, что для $n=4k+2$ кодов Баркера тоже не существует. Для некоторых значений $n=4k$ (8, 12, 16 и 20) путем полного перебора, который удалось немного сократить за счет одного установленного нами свойства, кодов Баркера тоже не было обнаружено. Полученные результаты были тоже опубликованы [17]. А спустя 6 лет нам прислал письмо сотрудник МТИ Р. Турун, который заинтересовался работами советских авторов и предлагал встретиться в Москве на международном математическом конгрессе. Однако это предложение поступило в наш институт спустя пару месяцев после окончания конгресса. Видимо, спецслужбы долго искали наш адрес в Горьком. К тому же в те времена несанкционированные встречи с иностранцами не приветствовались.

3. Трагическая судьба ТЭВМ

После ввода в эксплуатацию машины ГИФТИ, модернизации ее устройств ввода и вывода, а также устранения дефектов конструкции коллектив разработчиков, насчитывающий 6 квалифицированных электронщиков и порядка 10-12 монтажников оказался на перепутье. Его надо было чем-то занять, и, естественно, что новые задачи должны были соответствовать технологическим достижениям в стране, появились полупроводниковые приборы, пришедшие на смену радиолампам, ферритовые сердечники начали вытеснять прежние не очень оперативные запоминающие устройства (магнитные барабаны) и т.д.

И тогда по договору между ГИФТИ и Проектно-Технологическим Научно-Исследовательским Институтом Волго-Вятского Совнархоза (ПТНИИ ВВСНХ) было решено разработать ЭВМ нового поколения, получившую название ТЭВМ (Технологическая ЭВМ). Возник этот проект не спонтанно. Преподаватель кафедры теории колебаний А.М. Гильман, один из авторов проекта машины ГИФТИ, на протяжении нескольких лет разрабатывал программы, с помощью которых можно было бы автоматически формировать технологические карты для механической обработки деталей [18]. В дальнейшем подобные работы составили основу направления с аббревиатурой АСУ ТП (автоматизированные системы управления технологией производства). Однако эта

деятельность не очень вписывалась в научные направления, развиваемые в ГИФТИ. Поэтому А.М. Гильман устроился на работу по совместительству в ПТНИИ, где был создан соответствующий отдел, а возглавил его бывший сотрудник ГИФТИ Ю.А. Первин. Не без их поддержки руководство ПТНИИ обратилось с соответствующей просьбой к директору ГИФТИ Я. Н. Николаеву и заручилось его согласием.

Начальный период работы над проектом ТЭВМ протекал довольно спокойно. Были отработаны полупроводниковые аналоги основных блоков компьютера. А.М. Гончаров и А.С. Тарантович разработали специальный автомат для отбора ферритовых сердечников с одинаковыми характеристиками – элементов оперативной памяти. К прошивке ферритовых матриц, из которых собирался куб будущего запоминающего устройства (ЗУ), кроме монтажников и техников привлекались и другие сотрудники ВЦ. Мне удалось придумать схему прошивки координатной матрицы ЗУ, более устойчивую к разбросу физических параметров ферритовых сердечников. В стандартной схеме по координате x (младшие разряды адреса) и координате y (старшие разряды адреса) в обмотки ферритов координатной матрицы подаются два токовых импульса. Их сумма в единственном сердечнике координатной матрицы, находящемся на пересечении заданных координат, оказывалась достаточной для перемагничивания сердечника, управлявшего выбором указанной ячейки памяти. Во все остальные ферриты указанных столбцов и строк координатной матрицы поступали импульсы, мощность которых была вдвое меньше, чем это требовалось для перемагничивания. Поэтому они не возбуждались и не посылали в куб ЗУ сигнал чтения. Однако из-за разброса физических характеристик некоторые ферриты координатной матрицы могли сработать и при полутоковом импульсе. Поэтому инженеры старались разработать схему выборки, при которой в сердечники координатной матрицы подавались импульсы меньшей мощности, например, по $1/3$ тока перемагничивания. При этом в нужный сердечник подавались три таких импульса, а во все соседние поступал только один такой сигнал, который не мог их возбудить. Сложность заключалась только в том, чтобы придумать конструктивный способ прошивки ферритов координатной матрицы дополнительными сигнальными проводами. В работе [4] такой способ был предложен.

ТЭВМ представляла собой полупроводниковую машину последовательного действия, в которой было достигнуто довольно высокое быстродействие: за одну секунду она могла выполнить до 6000 логических операций, около 4000 сложений и порядка 260 умножений. Запятая была фиксирована между 18 и 19 разрядами, что позволяло оперировать с числами из диапазона $2^{-18} \leq |x| \leq 2^{29} - 2^{-18}$. Специфика системы команд ТЭВМ [19] заключалась, в первую очередь, в возможности удобной работы с фрагментами 48-битного слова. В слове можно было выделить до 8 групп смежных двоичных разрядов, в которых могла

находиться упакованная информация. Границы таких групп можно было либо набирать на пульте управления, либо изменять программным путем. Специальные команды ввода/вывода позволяли без какой-либо программной обработки вводить десятичные числовые данные в указанные группы и также автоматически переводить содержимое групп в десятичные числа при выводе. Система команд ТЭВМ включала довольно большой набор логических и арифметических операций над упакованными данными. Вторая немаловажная особенность ТЭВМ заключалась в использовании семи индексных регистров, с помощью которых можно было за один машинный такт не только модифицировать заданные адреса команды, но и соответствующим образом изменять содержимое указанного индексного регистра. Чтобы приблизительно оценить экономию от использования такого рода операций, было проведено сравнение технологических программ [20], разработанных по идентичным алгоритмам для ЭВМ Урал-4 (20000 машинных слов) и ТЭВМ (4000 машинных слов). Такая разница в объеме программ была достигнута и за счет развитой системы переадресации (в технологической программе использовалось до 30-35% команд с индексруемыми адресами).

Однако по мере приближения заключительного этапа, – межведомственных испытаний ТЭВМ, – отношения между заказчиком работ и их исполнителем постепенно портились. А.М. Гильман, считавший себя непревзойденным профессионалом в области архитектуры ЭВМ, пытался вносить все новые и новые изменения в конструкцию ТЭВМ. Коллектив разработчиков машины ГИФТИ, накопивший серьезный опыт при создании ЭВМ предыдущего поколения, со многими из этих изменений не соглашался. Возможно, что в ухудшении отношений сыграла свою роль и личностная неприязнь втянутых в конфликт людей. Может быть, А.М. Гильман хотел посчитаться с руководством ГИФТИ за то, что его идеи автоматизации технологических процессов не встретили взаимопонимания. Но, так или иначе, он приложил все усилия для того, чтобы руководство ПТНИИ отказалось от приемки ТЭВМ. Для директора ГИФТИ Я.Н. Николаева этот конфликт закончился потерей должности. Вместо эксплуатации более совершенной ТЭВМ, ей была уготована участь музейного экспоната (к сожалению, не сохранившегося до наших дней). Коллектив разработчиков распался – А.М. Гончаров ушел в качестве начальника ВЦ на завод радиодеталей, Г.Д. Зарницын перешел на должность начальника ВЦ в НИИ Радиосвязи, Р.Х. Садеков продолжил работу в роли начальника ЭВМ в ПТНИИ.

4. Серийная ЭВМ М-20

После того как лаборатория Н.А. Железцова почти полностью переключилась на решение своих задач с помощью цифровой ЭВМ, в ГИФТИ было при-

нято решение о приобретении самой мощной на тот период серийной ЭВМ М-20. Спонсорскую помощь как в получении наряда на очень дефицитное изделие, так и в финансовом плане оказал наш постоянный заказчик – ОКБМ (в те времена самые мощные компьютеры в первую очередь выделялись предприятиям, работающим на оборонную промышленность). В мае 1960 г. большая бригада инженеров и электромехаников во главе с Юрием Яковлевичем Хохловым выехала в Казань, где на заводе математических машин собирались и тестировались последние модели ЭВМ первого поколения – М-20. В составе бригады оказались и два программиста – Ю.Л. Кетков и В.А. Миронов. Нам предстояло ознакомиться с системой команд новой ЭВМ, ее программным обеспечением и тестовым хозяйством. Я получил дополнительное негласное поручение – свести к минимуму количество пустых блоков-заглушек путем замены их на сверхдефицитный ЗИП. Так как контроль за использованием блоков во время одновременной наладки очередной партии компьютеров на заводе был налажен из рук вон плохо, то к концу командировки в стойках нашей машины на местах заглушек оказался удвоенный комплект запасных блоков, очень пригодившихся нам во время эксплуатации М-20.

По тому времени М-20 была самой мощной ламповой ЭВМ первого поколения. Она имела оперативную память в 4096 45-разрядных ячеек и обладала средним быстродействием порядка 20 тысяч оп/сек. Числовые данные, обрабатываемые на М-20, были представлены в формате с плавающей запятой (1 бит – знак числа, 1 бит – знак порядка, 6 бит – двоичный порядок, 36 бит – двоичная мантисса). Машинная команда содержала 6-разрядный код операции, три 12-разрядных адреса и три битовых признака модификации адресов команды перед ее выполнением. Для модификации исполнительных адресов использовался единственный 12-разрядный индексный регистр (РА – регистр адреса). Содержимое РА можно было инициализировать, запоминать, увеличивать или уменьшать с помощью специальных команд, использовавшихся для организации циклов.

Единственная системная утилита, поставлявшаяся в комплекте машины, называлась «Интерпретирующей системой ИС-2». Она обеспечивала работу с библиотекой стандартных подпрограмм (СП), включавших средства преобразования числовых данных на вводе и выводе, процедуры вычисления элементарных функций и подпрограммы некоторых численных методов (решение систем линейных алгебраических уравнений, вычисление определенных интегралов, нахождение корней функций и др.). Разработка ИС-2 была выполнена в ОПМ под руководством М.Р. Шуры-Буры и при его непосредственном участии. Обращение к любой стандартной подпрограмме библиотеки ИС-2 выполнялось в две машинные команды:

k	016	k+1	7501	7610
k+1	nnn	x	N _{сн}	y

Первая команда выполняла безусловную передачу управления в начало ИС-2, располагавшуюся в памяти с ячейки 7501, она же запоминала адрес возврата (k+1) в ячейке 7610. Во второй строке (ячейка с адресом k+1) располагалась информация о входных (адрес x) и выходных (адрес y) данных, номер вызываемой подпрограммы (N_{сн}) и дополнительные сведения (код nnn), по разному использовавшиеся в разных СП. Подпрограмма, к которой происходило обращение в первый раз, вызывалась с магнитного барабана на специальное поле оперативной памяти. Это поле можно было расширить для того, чтобы на нем могло одновременно располагаться несколько наиболее часто используемых СП. После переполнения поле очищалось, и последующие обращения к СП снова вызывали коды нужных подпрограмм в оперативную память. Библиотека СП, в принципе, была открытой и позволяла пополнять свой состав пользовательскими подпрограммами, оформленными по установленным правилам. Первые годы программы для М-20 составлялись только в машинных кодах.

Установке М-20 в старом здании ГГУ (ул. Свердлова, 37) предшествовала довольно большая подготовительная работа по оборудованию мощной генераторной и холодильной установок, располагавшихся в подвальных помещениях. В машинном зале предстояло сделать фальшпол, под которым проходили воздуховоды с поступающим для охлаждения воздухом и размещалось кабельное хозяйство. М-20 потребляла порядка 200 кВт электроэнергии, и ее стойки нуждались в серьезном охлаждении. Система охлаждения была трехступенчатой – в шкафы с блоками снизу вдувался воздух с температурой 10-12 градусов, к верхней части шкафов температура воздуха уже доходила до 30-35 градусов. Затем нагретый воздух проходил через радиаторы с водой, которая, в свою очередь, охлаждалась мощной фреоновой установкой. Систему охлаждения удалось упростить, сведя ее к двухконтурному варианту – во дворе ГГУ пробурили артезианскую скважину, из которой зимой и летом поступала вода температурой в 8°.

Запуск М-20, который проводила пуско-наладочная бригада казанского завода математических машин, пришелся на 8 марта 1961 г. (тогда этот день еще относился к разряду рабочих). Начальником машины был назначен Г.Д. Зарницын. Круглосуточная работа М-20 обеспечивалась четырьмя сменами инженеров. В утренней и вечерней сменах дежурили более опытные инженеры – С.Г. Кузин, Э.Л. Иванова, Л.Ф. Полова и переквалифицировавшийся из программистов В.А. Миронов. Они довольно быстро научились находить и устранять неисправности в основных стойках М-20 (устройство управления, арифметическое устройство). Оперативная память находилась под опекой

Г.Д. Зарницына. За работу внешних устройств (магнитные барабаны, магнитные ленты, алфавитно-цифровое печатающее устройство) отвечала группа, возглавляемая Е.П. Цветковым. Полезное время, которое отдавалось программистам для решения задач, в лучшем случае, составляло 8-10 часов, как правило, падавших на вечернее и ночное время. Дневное время обычно уходило на профилактику и ремонт машины и ее периферийных устройств. Первая неприятность нас ожидала на майские праздники – предстояло три выходных, и машину решили выключить. Сразу после включения, которое по технологии «холодного старта» занимало примерно 6 часов, в ЭВМ полетело порядка 800 ламп. Хорошо, что у нас оказался двойной комплект ЗИП, но после этого события машину больше не выключали.

4.1. Первая электронная таблица УИС-РГМ

В 1963 г. в ВЦ ГИФТИ обратился главный инженер СКБ «ВолгоБалтСудопроект» А.А. Брайловский, который привел к нам иногородних заказчиков. Это были сотрудники отделения радиусографии НИИ Технологии Машиностроения (г. Ростов-на-Дону) Н.И. Курилов и В.Б. Перлин. По заказу СКБ в этом отделении разрабатывалась новая система подготовки плазовой документации для проекта нефтерудовоза, который должен был строиться на заводе «Красное Сормово». Метод радиусографии для проектирования судовой поверхности был придуман руководителем ростовского отделения Даниилом Самуиловичем Китаиновым. Идея этого метода заключается в следующем: некоторым эвристическим способом внутри корпуса судна и на его поверхности проводится несколько плавных линий, построенных из отрезков прямых и дуг окружностей. Эти линии условно называются радиусографическим ключом. Любое продольное (батоксы и ватерлинии) или поперечное (шпангоуты) сечение судовой поверхности, в свою очередь представляется как линия, образованная отрезками прямых и сопрягающимися с ними дугами окружностей. Центры этих дуг и точки сопряжения фрагментов сечений находятся на линиях ключа.

На всех судостроительных заводах страны в те времена использовалась старинная (чуть ли не со времен Петра Первого) технология формирования чертежной документации. В натуральную величину на полу специального цеха (плаза) строился чертеж поперечных сечений, содержащий 21 теоретический шпангоут. Размеры некоторых конструктивных линий были заданы разработчиками проекта. Плавные промежуточные сечения строились с использованием гибких реек (англ. термин – *spline*), которые прижимались к полу специальными грузиками в форме утюжков. Через точки шпангоутов, расположенные на определенной высоте, также с помощью гибких реек проводились ватерлинии. Точно такие же построения выполнялись и по третьей координате (по ширине судна). Этот процесс носил название «согласования» чертежа. В

дальнейшем полученные сечения использовались для образмеривания различных судовых деталей, определения размеров и профилей листов обшивки и др. Совершенно очевидно, что полученная таким образом чертежная документация для одного и того же проекта на разных судостроительных заводах, как правило, получалась разной. Поэтому при ремонте судна или установке на нем нового оборудования все приходилось делать только на заводе-изготовителе. На авиастроительных заводах тоже использовалась подобная технология, но чертежи сечений выполнялись в масштабе 1:10.

Основная идея радиусографического метода заключалась в попытке аналитически задать форму судовой поверхности и заменить вычерчивание и замеры сечений решением соответствующих геометрических задач и расчетом координат тех или иных точек. В отделении радиусографии НИИТМ была проделана довольно большая подготовительная работа – были созданы перечни стандартных графических задач, участвовавших в расчете различных сечений, подготовлены формуляры для записи последовательности вычислений и использования результатов решения каждой задачи в последующих построениях. Однако автоматизировать процесс обработки этих формуляров с помощью ЭВМ сотрудникам НИИТМ не удалось. Именно поэтому они обратились в ВЦ ГИФТИ с предложением о совместной работе.

Идея автоматизации подобного рода расчетов ассоциировалась у меня с обобщением логики машинных команд трехадресной машины М-20. Формат ее обычной команды выглядел следующим образом:

КОП А1 А2 А3

Здесь

КОП – код операции;

А1 – адрес первого операнда в оперативной памяти;

А2 – адрес второго операнда в оперативной памяти;

А3 – адрес ячейки, в которую засылается результат выполненной операции

Я подумал, а почему бы не смоделировать машину, приспособленную для решения геометрических задач, в которой каждая «команда» выполняла соответствующий графический расчет и имела бы следующий вид:

SOURCE NЗАД DEST

Здесь

SOURCE – вектор исходных данных;

NЗАД – номер решаемой задачи (аналог кода операции машинной команды);

DEST – вектор результатов

После анализа библиотек геометрических процедур было решено, что все вектора исходных данных содержат по 8 компонент, а все вектора результатов – по 3 компоненты. Фактическое количество параметров в некоторых задачах могло оказаться и меньше, тогда последние компоненты занимали место в памяти, но не использовались. Это позволяло организовать массивы исходных данных и массивы результатов в виде матриц с фиксированным количеством столбцов. Восьмеричная система, использовавшаяся на М-20 для нумерации ячеек оперативной памяти, очень удачно «ложилась» в выбранные размеры. Нумерация «ячеек» моделируемой графической ЭВМ могла восприниматься следующим образом. Первые три цифры определяли номер строки SOURCE (их можно было рассматривать как адрес «команды»), а младшая цифра соответствовала порядковому номеру исходного параметра (0 – первый параметр, 1 – второй, ..., 7 – восьмой). Для индикации результатов выполнения каждой графической «команды» было достаточно одной восьмеричной цифры, в которой использовались только три комбинации (4 – первый результат, 2 – второй результат, 1 – третий результат). Алфавитно-цифровых устройств подготовки данных в те времена на ЭВМ еще не было, поэтому пришлось ограничиться принятой цифровой индикацией строк и столбцов матрицы графической «программы».

Для организации логических операций в вычислительном процессе была введена группа дополнительных «задач», с помощью которых осуществлялись:

- проверка попадания переменных x , y , z в заданный диапазон и переход на соответствующую ветку графической программы;
- организация циклического повторения расчета с заданным шагом по той или иной координате;
- выдача тех или иных результатов решения на печать;
- рассылка полученных результатов по нужным клеткам матрицы исходных данных.

Оставалось только написать программу, которая бы эмулировала «графическую машину» на М-20. В таком эмуляторе была использована логика интерпретации вызовов стандартных программ – ИС-2. Библиотека графических задач была оформлена по правилам, принятым в ИС-2. Вызов нужной задачи осуществлялся по стандартной схеме. И это нашло свое отражение в названии УИС-РГМ (Универсальная интерпретирующая система радиусографического метода).

Процедура «программирования» сводилась к переписи таблиц НИИТМ на бланках машинных команд М-20 и могла выполняться неквалифицированным персоналом. Первая версия УИС-РГМ заработала в конце 1963 г., и по ее материалам я написал свою первую монографию (точнее, руководство системного программиста и руководство пользователя), которая была опубликована

в Ростове-на-Дону под грифом «секретно». На данном грифе настояли наши заказчики, которые вели ряд проектов для оборонных предприятий (КБ Миля, КБ Антонова, СКБ Лазурит и др.).

Разработанный подход я назвал «матричным программированием». По сути дела, его идеология послужила прототипом для современных электронных таблиц.

5. НИИ ПМК и БЭСМ-3М

Осенью 1964 г. при ГГУ вскоре после открытия факультета ВМК был создан НИИ Прикладной математики и кибернетики, первым директором которого стала Людмила Николаевна Беллюстина. Наиболее важная роль в создании НИИ ПМК, наверное, принадлежит Юрию Исааковичу Неймарку, Евгении Александровне Леонтович-Андроновой и Аркадию Моисеевичу Гильману. Основную часть штата нового института составили сотрудники ГИФТИ, перешедшие в НИИ ПМК вслед за своими лидерами. Самый крупный отдел института (отдел №2) возглавил Ю.И. Неймарк. Однако в НИИ ПМК появились и новые подразделения, которых ранее в ГИФТИ не было – отдел №3, нацелившийся на решение задач управления производством и технологическими процессами (его возглавил к.т.н. А.М. Гильман), отдел №4, первоначально ориентированный на численное решение задач математической физики (его руководителем стал к.ф.-м.н. Соломон Иосифович Альбер), лаборатория бионики и биокибернетики (ее возглавил к.б.н. М.Ю. Ульянов). Целиком перешла в НИИ ПМК группа, занимавшаяся компьютерной лингвистикой и возглавляемая Виктором Алексеевичем Аграевым. На ее базе в НИИ ПМК была создана лаборатория №8. Отделом №5, в задачи которого входила эксплуатация средств вычислительной техники, руководил Юрий Яковлевич Хохлов, ранее работавший главным инженером ВЦ ГИФТИ.

В связи с приобретением новой полупроводниковой ЭВМ БЭСМ-3м, совместимой по системе команд с компьютерами типа М-20, и появлением в научных подразделениях НИИ ПМК довольно большого контингента непрофессиональных пользователей, руководство института решило организовать при отделе №4 лабораторию автоматизации программирования. Самым простым способом для решения этой задачи было переманивание ведущих сотрудников ВЦ ГИФТИ, которые уже несколько лет занимались эксплуатацией ЭВМ с идентичной системой команд и с таким же набором обслуживающих программ. Зарплата заведующего лабораторией без степени меня не очень прельщала. В денежном исчислении добавка к зарплате была очень небольшой. В вычислительном центре моя штатная должность к тому времени достигла потолка – я числился главным инженером–конструктором, но перспектив дальнейшего ро-

ста без защиты кандидатской диссертации в ГИФТИ не было. А напористый С.И. Альбер искушал меня именно этим. После трехмесячных переговоров я сдался – 1 апреля 1965 г. в штате 4-го отдела появились три новых программиста – Ю.Л. Кетков, Т.С. Ямпольская и Н.М. Станкевич. Одновременно с нами из ВЦ ГИФТИ в штат 4-го отдела была переведена лаборантка М.Л. Белова.

Начальная деятельность в новой лаборатории сводилась к освоению новых систем программирования – транслятора с алгоритмического языка АЛГОЛ-60 (ТА-2М), автокода БМ-4/220, пакета графических программ ГРАФОР, расширения интерпретирующей системы ИС-22. Приходилось обучать работе с этими средствами программистов из других подразделений, оказывать им консультации, помогать в поиске ошибок и отладке программ. Появление в городе ЭВМ, совместимых с М-20, расширило круг наших заказчиков. За аналогичной помощью к нам обращались сотрудники ГКТ БИП (ныне НИИ Измерительных Систем им. Ю.Е. Седакова), ОКБМ, Политехнического института. Одновременно с этой деятельностью мы довольно активно занимались разработкой новых стандартных программ для ИС-22. Несколько десятков таких СП было зарегистрировано в Государственном фонде алгоритмов и программ.

Вторым направлением нашей лаборатории было создание отдельных программ и систем для автоматизации проектных и конструкторских работ в судостроении и авиастроении. Пригодились старые связи, накопленные на стадии внедрения радиосографического метода расчета сечений судовой поверхности и развертки листового материала, необходимого для обшивки судов. В числе наших заказчиков оказались ведущие судостроительные КБ г. Горького – СКБ Волго-балтсудопроект, ЦКБ Лазурит, СКБ по судам на подводных крыльях. Кроме них аналогичными услугами пользовался ряд иногородних предприятий судостроительной и авиационной промышленности – Николаевский судостроительный завод, ЦНИИ им. акад. Крылова (Ленинград), КБ Миля (Ростов-на Дону), КБ Ильюшина (Москва), п/я 202 (Новосибирск), КБ Антонова (Киев).

5.1. Линейные сплайны

Решение многих прикладных задач связано с аппроксимацией различных кривых – нелинейных характеристик систем автоматического регулирования, траекторий режущего инструмента на станках с числовым программным управлением, контуров сложных криволинейных поверхностей в судостроении и авиастроении и т.п.

Если нужно построить единственный отрезок прямой, наилучшим образом приближающий на отрезке $[a, b]$ заданную кривую, то эта задача элементарно решается с помощью метода наименьших квадратов. И сводится она к решению системы линейных алгебраических уравнений второго или третьего по-

рядков в зависимости от способа задания прямой $-y = k \cdot x + b$ или $A \cdot x + B \cdot y + C = 0$. Однако как только на заданном отрезке $[a, b]$ нужно построить два звена оптимальной ломаной, то сразу появляется новый нелинейный параметр (точка излома) и дополнительные ограничения, связанные со стыковкой звеньев ломаной. И это резко осложняет задачу.

Еще сложнее построить ломаную с неизвестным числом звеньев, каждое из которых аппроксимирует исходную кривую с требуемой точностью. Можно показать, что построение очередного звена ломаной эквивалентно ответу на вопрос – как далеко можно шагнуть вдоль кривой, чтобы стрелка прогиба построенной хорды была равна заданной точности. Конечно, для этой цели можно построить программу, которая постепенно увеличивая шаг одновременно вычисляла бы и стрелку прогиба очередного приближения. Но хотелось бы найти аналитическое решение, исключавшее такой перебор и позволявшее оценить количество звеньев оптимальной ломаной.

В конце 1965 г. мне это удалось – я построил приближенные формулы для вычисления квазиоптимального шага h по значениям производных исходной функции и заданной точности аппроксимации ε :

$$h_1 = \sqrt{\frac{8\varepsilon \sqrt{1 + [f'(x_n)]^2}}{|f'(x_n)|}} \quad \text{для явного задания кривой } y=f(x)$$

$$h_2 = \sqrt{\frac{8\varepsilon \sqrt{(x'_n)^2 + (y'_n)^2}}{|x'_n \cdot y''_n - x''_n \cdot y'_n|}} \quad \text{для параметрического задания кривой } x=x(t), y=y(t)$$

Если в качестве параметра t выбирается длина дуги s (лонгальный параметр), то имеет место:

$$h_2 = \sqrt{\frac{8\varepsilon}{|K(s_n)|}} \quad \text{где } K(s_n) \text{ – кривизна в точке } x(s_n), y(s_n).$$

Удалось доказать, что на участках кривых с монотонным изменением кривизны аппроксимация «вперед» ($x_{n+1} = x_n + h$) всегда позволяет строить звенья, которые отклоняются от кривой чуть больше, чем на ε . А при аппроксимации «назад» ($x_{n+1} = x_n - h$) все звенья ломаной уклоняются от кривой чуть меньше, чем на ε . Таким образом удалось получить верхнюю и нижнюю границы для оценки количества звеньев оптимальной ломаной. Для кривых второго порядка, чаще других используемых при конструировании судовых поверхностей, эти оценки оказались справедливыми не только для малых значений погрешности.

Ю.С. Завьялов (ИМ СО АН СССР), составлявший антологию работ по сплайнам в СССР и за рубежом, отнес этот результат к числу первых конструктивных достижений в области линейных сплайнов [21].

Конец 1965 и начало 1966 г. у меня ассоциируются с защитой кандидатской диссертации на соискание ученой степени кандидата физико-математических наук по теме «Об оптимальных методах нелинейной аппроксимации плоских кривых и системе автоматизации программирования для обработки геометрической информации». К форсированию этого рубежа много усилий приложил мой непосредственный начальник Соломон Иосифович Альбер, согласившийся стать моим научным руководителем. И хотя к результатам, включенным в тему диссертационной работы, он прямого отношения не имел, определенная заинтересованность у него была. Во-первых, я становился его первым кандидатом, во-вторых, защита повышала престиж нашего отдела и его заведующего, в-третьих, он активно отстаивал научную новизну полученных результатов и защита могла стать тому подтверждением. Так или иначе, под его напором в 1966 г. были подготовлены две основополагающие статьи [22, 23] и на ученом совете Горьковского университета состоялась защита диссертации. В качестве первого оппонента выступил профессор Ю.И. Неймарк. По первой части работы он имел хорошее представление и достаточно высоко оценил ее. Но программная часть диссертации первому оппоненту была неподвластна. И тут сказалась квалификация второго оппонента – первого кандидата наук по системному программированию в нашей стране Э.З. Любимского. Он был приглашен из ОПМ и второй раз выступил в роли моего опекуна. Тем более, что прототип универсальных интерпретирующих программ для ЭВМ типа М-20 был знаком ему не понаслышке – Э.З. Любимский был непосредственным участником создания ИС-2 (22). С единственным голосом против Ученый Совет проголосовал за присуждение искомой ученой степени. До меня доходили слухи о том, что несколько заинтересованных лиц пытались выяснить фамилию проголосовавшего против. Но на меня этот момент никакого впечатления не произвел – я предпочитал не вмешиваться в околонуточные споры, характерные для того времени. Да и кто бы тогда считался с моим мнением – моя весовая категория находилась в самом низу карьеры научной деятельности.

6. Первое знакомство с БЭСМ-6. ОС ИПМ

В середине 1967 г. Президиум Академии Наук СССР, которая тогда возглавлялась директором ИПМ академиком М.В. Келдышем, принял закрытое постановление о создании современного программного обеспечения флагмана нашей вычислительной техники – ЭВМ БЭСМ-6. Дело в том, что появившаяся в 1966 г. БЭСМ-6 была оснащена наспех созданной операционной системой –

диспетчером Д-67 (авторы Л.Н. Королев, А.Н. Томилин и др.) и упрощенной версией транслятора с автокода (автор М.Г. Чайковский). В рамках нового проекта предполагалось создание многопользовательской операционной системы, функционально не уступающей по возможностям самой мощной зарубежной ОС Multics (разработка Массачусетского технологического института), и серией трансляторов с современных алгоритмических языков. К этому моменту в отделе автоматизации программирования ИПМ, возглавляемому М.Р. Шурой-Бурой, уже были начаты некоторые разработки. В стадии запуска находился транслятор с Автокода БЕМШ, разрабатывавшийся в секторе В.С. Штаркмана, во всю шли работы над проектом алгоритмического машинно-ориентированного языка АЛМО (С.С. Камынин, Э.З. Любимский) и трансляторов других языков в код АЛМО. В соответствии с принятым постановлением ОПМ мог привлекать к выполнению указанных работ лучшие программистские силы из любых организаций страны, в т.ч. из ядерных центров Сарова и Челябинска, из ведущих НИИ оборонной и космической промышленности.

Моя недавняя защита кандидатской диссертации, упрочившая связи с ОПМ, позволила С.И. Альберу обратиться к руководству ИПМ с предложением помощи, в результате чего двое сотрудников НИИ ПМК Ю.Л. Кетков и Н.М. Салганская в конце 1967 г. были отправлены в длительную командировку в Москву. Нина Салганская удачно вписалась в коллектив трансляторщиков, создававших свою продукцию на базе языка АЛМО, а я попал в уже знакомый мне коллектив отдела, возглавивший работу над проектом операционной системы ОС ИПМ. В начальный момент в головную группу от ИПМ входили С.С. Камынин, Э.З. Любимский, И.Б. Задыхайло и В.С. Штаркман. От сторонних организаций в состав этой группы были включены представитель НИИ-4 майор Д.А. Корягин и представитель НИИ ПМК Ю.Л. Кетков. На роли лидеров претендовали В.С. Штаркман, имевший к тому времени опыт создания операционной системы ЭВМ «Весна», и тандем ЛЮКС (Любимский – Камынин). Однако видение архитектуры ОС у претендентов оказалось различным, и вскоре В.С. Штаркман отошел от работы в проекте ОС ИПМ, ограничившись совершенствованием Автокода БЕМШ. Он же принимал участие в создании драйверов некоторых внешних устройств.

После некоторого притирания друг к другу, функции в головной группе разделились следующим образом: С.С. Камынин – генератор идей и неременный участник обсуждения текущей ситуации; Э.З. Любимский – фактический руководитель работ по созданию ОС ИПМ и проекта АЛМО; И.Б. Задыхайло и Д.А. Корягин – ответственные исполнители программ управления периферийными устройствами (в нашем проекте эта сфера получила название «супервайзер»); Ю.Л. Кетков – управление исполнителями операционной системы, заданиями, процессами и их ресурсами (эта сфера называлась «монитором»).

6. Первое знакомство с БЭСМ-6. ОС ИПМ

Работы над основными компонентами ОС ИПМ продолжались более года. Члены головной группы решили отпускать усы и/или бороды и дали обет не сбривать свою растительность до тех пор, пока ОС ИПМ не будет сдана в эксплуатацию. Появление усатой команды в столовой ИПМ неизменно сопровождалось взрывами смеха и колкими замечаниями в адрес разработчиков программного обеспечения БЭСМ-6. В 1970 г. началась опытная эксплуатация ОС ИПМ, которая была передана в ВЦ СО АН СССР. В издательском отделе ИПМ была выпущена серия препринтов, посвященных отдельным компонентам системы (один из них содержит общее описание монитора – [24]). Несколько докладов по проделанной работе (в т.ч. и с моим участием – [25,26]) были представлены на Второй Всесоюзной конференции по программированию в Новосибирске (1970 г.). По материалам ОС ИПМ в 1971 г. была подготовлена докторская диссертация Э.З. Любимского на тему «Возможности и принципы построения операционной системы для БЭСМ-6 (ОС ИПМ)», которая спустя 2 года была защищена в Институте Кибернетики АН УССР.



Слева направо: Э.З. Любимский, А.Д. Корягин, Ю.Л. Кетков, И.Б. Задыхайло, Л.В. Ухов
(у входа в ИПМ АН СССР)

7. Появление Бейсика в нашей стране

Во время разработки ОС ИПМ нам очень не хватало режима диалога для отладки компонент системы. Несколько раз мы обсуждали идею многотерминального диалогового монитора, который позволял бы предоставить пользователям БЭСМ-6 оперативный доступ к своим программам. Впоследствии такой монитор под названием ДИМОН был разработан одним из членов нашей команды С.А. Усовым. Однажды Э.З. Любимский принес на работу руководство пользователя по интерпретирующей диалоговой БЭЙСИК-системе для ЭВМ фирмы General Electric (GE-400). У него возникла идея реализовать нечто похожее в рамках ОС ИПМ, но эту идею временно пришлось отложить. По руководству я познакомился с операторами входного языка BASIC (аббревиатура от **Beginner All Purpose Symbolic Instruction Code** – **многоцелевой язык символических инструкций для начинающих**). Бэйсик-система мне понравилась больше, чем система разделения времени на базе языка JOSS (сообщение об этой системе на одном из наших семинаров сделал сотрудник ВЦ АН СССР В.М. Курочкин).

Возвратившись после полугодовой командировки в Горький, я предложил сотрудникам своей лаборатории реализовать пошаговый компилятор с языка Бейсик. У нас еще не было такого большого проекта, к выполнению которого можно было подключить несколько человек. К пяти штатным сотрудникам отдела (М.М. Белослудцевой, Э.Н. Ильину, Ю.Л. Кеткову, И.М. Куракиной и В.С. Максиму) присоединился еще один энтузиаст из аналогичного подразделения НИРФИ – В.В. Бочаров. И до конца 1969 г. нам удалось разработать пошаговый компилятор с языка Бейсик в код машины типа М-20. В рамках системы Горьковского университета к тому времени уже эксплуатировались две такие ЭВМ – БЭСМ-3М в НИИ ПМК и М-222 в ВЦ ГИФТИ.

При создании компилятора мы преследовали следующие цели. Во-первых, будущее его использование должно было максимально просто обеспечить режим разделения времени для поддержки одновременной работы нескольких терминалов. Для этого активному заданию временно выделялся основной куб памяти (МОЗУ-0 емкостью 4К 45-разрядных слов), где по фиксированной схеме распределения оперативной памяти располагались откомпилированная программа и ее исходные данные, интерпретирующая система ИС-2 и рабочее поле для ее стандартных подпрограмм. Второй куб (МОЗУ-1) использовался для работы компилятора и хранения управляющих таблиц, описывающих состояние каждого пользователя системы. Переключение на обслуживание очередного абонента предполагало смену содержимого единственного базового регистра.

Во-вторых, на магнитном барабане предполагалось хранить образы кубов оперативной памяти, соответствующие обслуживаемым пользователям. На

ЭВМ типа М-222, располагавших 8 кубами оперативной памяти, сразу несколько готовых к исполнению программ могли находиться в оперативной памяти одновременно (это существенно экономило время свопинга).

Перед выполнением группы команд, соответствовавших очередному оператору активной программы, порядковый номер этого оператора фиксировался в виртуальном «счетчике команд», что позволяло идентифицировать ошибки периода выполнения с соответствующей строкой исходной программы. Управляющая программа по таймеру компьютера могла переключать внимание компилятора на обслуживание активных абонентов системы по кругу с полным периодом порядка 1-1.5 сек. Это позволяло практически мгновенно отреагировать на оператор, введенный с терминала пользователя, и либо превратить его в соответствующий эквивалент в машинных кодах, либо выдать сообщение о допущенной ошибке. Во время отсутствия завершенных сообщений с терминалов по фиксированным квантам времени осуществлялись запуски готовых к исполнению программ. Объем исходной программы мог достигать до 300 операторов, чего было вполне достаточно на начальной стадии обучения. Архив исходных программ располагался на магнитной ленте и мог одновременно хранить тексты до 64 программ на Бейсике.

Отмечу довольно важную особенность разработанного компилятора. Наряду с максимальным использованием ИС-2 для вычисления элементарных функций языка Бейсик, мы включили во входной язык оператор БЭСМ, позволяющий вставлять в текст исходной программы машинные команды с символическими адресами. По сути дела, эту идею мы заимствовали у транслятора с Алгола ТА-2М, где была реализована процедура М20-Л. **Основным назначением** оператора БЭСМ, конечно, была возможность подключения довольно мощной библиотеки стандартных программ ИС-2/22. Но можно посмотреть и шире – в современных системах программирования (например, в компиляторах языка Си) возможность ассемблерных вставок реализуется с помощью оператора `asm`. Это ничем не отличается от оператора БЭСМ и его прототипа М20-Л, **которые были придуманы и реализованы задолго до появления аналогичных средств** в сегодняшних алгоритмических языках высокого уровня.

Пошаговый компилятор с Бейсика был написан на автокоде БМ-4/220. В НИИ ПМК он эксплуатировался в пакетном режиме [29-31] с вводом исходной Бейсик-программы на перфокартах. При этом пропускная способность на студенческих задачах в 10-15 раз превосходила аналогичные показатели для системы программирования на Фортране (комплекс Ф-20). С небольшими изменениями схемы переключения пользователей и подключением драйверов электрифицированных пишущих машинок типа «Консул-254» компилятор активно использовался в ВЦ ГИФТИ в диалоговом режиме. Позднее шумящие терминалы были заменены на дисплеи типа «Видеотон-340». Преобразование

пакетной версии Бейсик-системы в многопультную систему коллективного пользования выполнили сотрудники ВЦ Д.М. Штейман, Л.Я. Дутышева и В.Г. Манишин. Диалоговый вариант получил название «Студент» [32, 33].

Оба варианта Бейсик системы привлекли широкое внимание со стороны учебных заведений и проектно-конструкторских организаций. Распространялись они на безвозмездной основе и были переданы более чем 150 организациям. Система «Студент» эксплуатировалась в ВЦ ГИФТИ до начала 80-х годов прошлого столетия. После серии публикаций [36-38] была подготовлена первая монография по языку Бейсик на русском языке, которую в 1978 г. опубликовало издательство «Финансы и статистика» [39]. Она могла бы появиться много раньше, но рукопись почти два года пролежала без движения в портфеле М.Р. Шуры-Буры, который курировал в этом издательстве раздел программного обеспечения ЭВМ.

Среди других организаций, заинтересовавшихся реализацией Бейсика на отечественных ЭВМ второго поколения, отмечу ВЦ СО АН СССР и Институт математики АН БССР. В Новосибирске группа энтузиастов, возглавляемая П.К. Леоновым и Г.И. Сердюком, в 1972 г. разработала проект диалоговой системы BASIC-6, которая должна была функционировать на БЭСМ-6 под управлением ОС ИПМ и обслуживать одновременно до 10 терминалов. Кроме доклада на конференции по «Теории языков и методы построения систем программирования» [34] и более поздней публикации в журнале «Кибернетика» [35] о судьбе системы BASIC-6 ничего не известно. Скорее всего, она была предана забвению, как и многие другие разработки, не доведенные до сколь-нибудь широкого внедрения. Примерно такая же судьба постигла и систему Бэйсик-ГАММА, разработанную в 1974 г. для ЭВМ Минск-32.

8. В кильватере IBM/360/370. ЕС ЭВМ и М-4030

Если большинство ЭВМ второго поколения у нас и за рубежом, выпускавшиеся разными производителями, были не совместимы между собой ни по системам команд, ни по периферийному оборудованию, то первый удар по сложившейся ситуации нанесла фирма IBM. В 1966 году она объявила о проекте выпуска серии вычислительных машин разной производительности с моделями, совместимыми от младших к последующим в аппаратных и программных решениях. Первая серия получила название семейства **IBM/360**, вслед за которым с небольшим интервалом по времени было начато производство более совершенного семейства **IBM/370**. Ряд менее мощных зарубежных производителей выстроились вслед за IBM и включили в свою программу выпуск ЭВМ со сходной архитектурой (английская ICL, немецкая Siemens). Экономический выигрыш для догоняющих лидирующую корпорацию заключался в возможно-

сти использования типового периферийного оборудования и оснащения своих ЭВМ программами, разработанными для IBM/360/370. В тот момент примерно 70% зарубежного рынка ЭВМ был представлено моделями IBM.

Аналогичное решение было принято и в нашей стране. Но с выпуском ЕС ЭВМ мы запоздали примерно на 6-7 лет (первые модели появились в нашей стране в 1972 году). Во-вторых, решение о копировании зарубежных моделей ЭВМ в двух ведущих министерствах (МРП – министерство радиопромышленности и МЭП – министерство электронной промышленности) практически похоронило лучшие отечественные разработки. Конечно, такой политический шаг имел и положительные моменты. Нам удалось (правда, с очень большим запозданием и на довольно устаревшей технологической основе) наладить выпуск электронных компонент (широкую номенклатуру интегральных схем, многослойные печатные платы и др.) и совместимого с зарубежными моделями периферийного оборудования (однако крайне низкого качества). Без приобретения каких-либо лицензий мы раздобыли минимально необходимое программное обеспечение – операционные системы и соответствующий набор компиляторов. Однако даже сейчас, спустя почти 40 лет, довольно трудно дать взвешенную оценку наших прошлых решений.

На первых порах такая серьезная фирма как IBM не обратила внимания на диалоговые возможности Бейсика. В рамках первых операционных систем MS-DOS преобладал пакетный режим обработки программ пользователей. Поэтому раньше других ЭВМ единой системы Бейсик появился на самой крошечной модели ЕС-1010, выпускавшейся в Венгрии. Но эта модель настолько выходила вон из ряда ЕС ЭВМ, что ни о каком переносе соответствующего Бейсик-интерпретатора речи не было.

8.1. Продолжение Бейсик-эпопеи

А у нас уже был накоплен опыт разработки и эксплуатации пошагового Бейсик-компилятора. И мы решили повторить эксперимент с созданием аналогичной версии, работающей под управлением MS-DOS. На первую версию, создававшуюся на БЭСМ-3М с нуля, мы потратили порядка 5-6 человеко-лет. И у нас уже были наработаны около полусотни ассемблерных макрокоманд, существенно упрощавших написание компилятора. Оставалось только перевести их в код новой ЭВМ и усовершенствовать схему управления. Почти тем же коллективом мы справились с этой работой за полгода и поделились своим опытом на двух Всесоюзных конференциях [40, 41]. Я думаю, что мы действовали по той же схеме, что и Дж. Томпсон с Денизом Ритчи, когда они перенесли операционную систему Unix с PDP-7 на PDP-11. Первая версия языка Си многим обязана набору макрокоманд, которые были накоплены в Bell Laboratories.

С появлением дисплейных станций типа ЕС-7920 мы перевели пакетную версию в режим диалога, и на протяжении нескольких лет в НИИ ПМК имела возможность удаленной оперативной работы на Бейсике под управлением MS-DOS.

8.2. Диалог на ФОРТРАНЕ с ЕС ЭВМ

Появление ЕС ЭВМ привело к практически полному отказу от программирования на АЛГОЛЕ, который долгое время оставался «государственным» языком советских программистов. На смену пришел Фортран, язык еще более древний по происхождению, но получивший гораздо более широкое распространение благодаря поддержке со стороны фирмы IBM.

Наш институт довольно активно занимался разработкой программного обеспечения для решения проектно-конструкторских задач на крупнейших в области предприятиях авиационной промышленности. На заводе Гидромаш проектировали и изготавливали шасси для боевых и гражданских самолетов. СКБ «Восход» и Павловский машиностроительный завод были ведущими организациями в стране по проектированию и выпуску узлов гидравлики, использовавшихся на всех изделиях оборонной и космической промышленности. С решением задач, требовавших применения различного рода графопостроителей, мы справились довольно быстро. Помогло то, что еще на машинах типа М-20 мы достаточно хорошо освоили возможности одного из первых графических пакетов ГРАФОР [42]. Поэтому задолго до того, как авторы пакета «вспомнили» о наступлении ЕС ЭВМ, мы с В.С. Максимовым адаптировали этот пакет для Фортрана ЕС ЭВМ [43] и начали применять его не только в НИИ ПМК, но и в сторонних организациях.

Однако при всех преимуществах Фортрана IV, реализованного в составе системного программного обеспечения MS-DOS, в этом языке был существенный минус – он был совсем не приспособлен для организации работы в диалоговом режиме. В нем отсутствовала возможность общения прикладной программы с пультом пользователя. Автоматическое распределение памяти, выполняемое транслятором, лишало пользователя возможности проанализировать значения данных по их естественным обозначениям во время выполнения программы или изменить значения некоторых управляющих параметров. Фортран ЕС не располагал гибкими возможностями для работы с символьными данными, не умел анализировать запросы проектировщика на естественном языке, не мог передать управление пользователю для выбора оптимального продолжения схемы проектирования и т.д.

Для преодоления всех этих препятствий был разработан пакет ДИАФОР (от ДИАлог на ФОРтрane) [44-46], который широко использовался для создания

ряда подсистем САПР, эксплуатировавшихся в СКБ заводов Горького и Павлово.

Далее приводятся названия глав из текста мемуаров Ю.Л. Кеткова, сами главы написаны не были - *примеч. ред.*

9. Второе свидание с БЭСМ-6. «Чертеж»

10. Выход в космос («Луна»)

11. IBM-совместимые персоналки

11.1. Визуализация цифровых карт

11.2. Шаблоны печатных плат

11.3. Смена кресла по основной работе – профессорство

11.4. Переход в следующую весовую категорию (защита докторской диссертации по совокупности работ)

11.5. Чукча, однако, – писатель. И его сын тоже

11.6. Увлечение MATLAB'ом

12. Олимпиады по информатике. Даешь свободное ПО

Литература

1. Камынин С.С., Любимский Э.З., Шура-Бура М.Р. Об автоматизации программирования с помощью программирующей программы. // Проблемы кибернетики, вып.1, 1958
2. Мартынюк В.В. О методе символических адресов. // Проблемы кибернетики, вып. 6, 1961
3. Кетков Ю.Л. Схема сокращенного деления на ЦВМ последовательного действия. // Известия вузов Радиофизика т.2, N 2, 1959
4. Кетков Ю.Л. Прошивка координатной матрицы ЗУ на магнитных сердечниках, работающей по принципу многократного совпадения токов. // Известия вузов Радиофизика т.4, N 1, 1961
5. Кетков Ю.Л. Об одном способе вычисления полиномов на математических машинах. //Известия вузов. Радиофизика, т.4, № 1, 1958 (дата поступления в редакцию 5.05.1958)
6. Белага Э.Г. Некоторые вопросы вычисления многочленов. //ДАН, 123, № 5, 1958 (дата поступления в редакцию 13.06.1958)
7. Пан В.Я. Некоторые схемы вычисления многочленов с вещественными коэффициентами. //ДАН, 127. №2, 1959 (дата поступления в редакцию 13.03.1959)
8. Белага Э.Г. О вычислении значений многочленов от одного переменного с предварительной обработкой коэффициентов. //Проблемы кибернетики, вып. 5, 1961 (дата поступления в редакцию 30.05.1958)
9. Пан В.Я. Некоторые схемы вычисления многочленов с вещественными коэффициентами. //Проблемы кибернетики, вып. 5, 1961 (дата поступления в редакцию 28.11.1958)
10. Пан В.Я. Вычисление многочленов по схемам с предварительной обработкой коэффициентов и программа нахождения параметров. //ЖВМ и МФ, 2:1, 1962
11. Пан В.Я. О вычислении многочленов пятой и седьмой степени с вещественными коэффициентами. //ЖВМ и МФ, 5:1, 1965
12. Пан В.Я. О способах вычисления значений многочленов. //УМН, 21:1, 1966
13. Пан В.Я. Об одновременном вычислении значений нескольких многочленов малых степеней (от двух до пяти). //ЖВМ и МФ, 6:2, 1966
14. Баталова З.С. Производство операций с плавающей запятой на машине с фиксированной запятой. // Известия вузов. Радиофизика, т.2, №5, 1959.
15. Кетков Ю.Л. Подпрограммы плавающей запятой. // Известия вузов. Радиофизика, т.3, № 1, 1960
16. Поляк Ю.В., Мошетов Р.В. Научные труды РТИ АН СССР, т.1, № 124, 1959.

17. И.М. Иванова, Ю.Л. Кетков, Т.С. Ямпольская О существовании кодов Баркера. // Известия вузов. Радиофизика, т.3, 1960
18. Гильман А.М. Проектирование технологических карт механической обработки на электронно-вычислительных машинах. ГОСИНТИ. М., 1959
19. Кетков Ю.Л., Жеглова Н.В. Система операций технологической машины. // Труды по вопросам применения электронных вычислительных машин в народном хозяйстве. – Горький, 1964
20. Антонова В.М., Баталова З.С., Кетков Ю.Л. Некоторые особенности программирования задачи проектирования технологических карт на машине ТЭВМ. // Там же
21. Завьялов Ю.С., Квасов Б.И., Мирошниченко В.Л. Методы сплайн-функций. – М.: Наука, 1980.
22. Кетков Ю.Л. Об оптимальных методах кусочно-линейной аппроксимации. // Известия вузов. Радиофизика, т.9, №5, 1966
23. Кетков Ю.Л. Автоматизация проектирования поверхностей корпусов судов. // Сб. статей «Автоматизация технологического проектирования при помощи ЭВМ». М.: Машиностроение, 1966
24. Кетков Ю.Л. Об оптимальных методах нелинейной аппроксимации плоских кривых и системе автоматизации программирования для обработки геометрической информации. – Автореферат диссертации на соискание ученой степени кандидата физико-математических наук. Горький, 1966. 8 с.
25. Кетков Ю.Л. О приближенных методах кусочно-линейной аппроксимации плоских кривых. – В сб. «Ученые записки. Прикладная математика и кибернетика. Труды НИИ ПМК и факультета ВМК ГГУ». – Горький, 1967
26. Задыхайло И.Б., Камынин С.С., Кетков Ю.Л., Корягин Д.А., Любимский Э.З., Штаркман В.С. Математическое обеспечение. Операционная система БЭСМ-6. Монитор. Препринт ИПМ АН СССР, М., 1969. 24 с.
27. Задыхайло И.Б., Камынин С.С., Кетков Ю.Л., Корягин Д.А., Любимский Э.З., Шура-Бура М.Р., Штаркман В.С. Монитор ОС ИПМ. – Труды ВКП-2, Новосибирск, 1970
28. Елтаренко А.Н., Задыхайло И.Б., Зеленина Л.А., Зусман И.Х., Камынин С.С., Кетков Ю.Л., Любимский Э.З., Шелепов Н.А. Управление задачами и процессами в ОС ИПМ. – Труды ВКП-2, Новосибирск, 1970
29. Белослудцева М.М., Бочаров В.В., Ильин Э.Н., Кетков Ю.Л., Куракина И.М., Максимов В.С. BASIC-система пакетной обработки задач на ЭВМ типа БЭСМ-4, М-220. Инструкция по эксплуатации (руководство). – Горький, 1971

30. Белослудцева М.М., Бочаров В.В., Ильин Э.Н., Кетков Ю.Л., Куракина И.М., Максимов. Использование алгоритмического разговорного языка BASIC на ЭВМ типа БЭСМ-4. // Труды II Всесоюзной конференции по применению ЭВМ и математических методов планирования и управления предприятиями. Горький, 1971.
31. Белослудцева М.М., Бочаров В.В., Ильин Э.Н., Кетков Ю.Л., Куракина И.М., Максимов В.С. BASIC-система пакетной обработки задач на ЭВМ типа БЭСМ-4, М-220, М-222. Учебное пособие. – Горький: изд. ГГУ, 1973. – 192 с
32. Дутьшева Л.Я., Манишин В.Г., Штейман Д.М. Совместимая система разделения времени «Студент» // УСиМ, 1975, №1.
33. Дутьшева Л.Я., Кетков Ю.Л., Кузин С.Г. Обучение программированию в СКП «Студент». – В сб. «Машинное обучение с помощью диалога». – Москва: МДНТП, 1976. – с.95-99
34. Леонов П.К., Сердюк Г.И. Язык и архитектура системы BASIC-6 // Труды симпозиума. Теория языков и методы построения систем программирования. Киев – Алушта, Институт кибернетики АН СССР, 1972. – с.228-243.
35. Головлева Н.В., Леонов П.К., Сердюк Г.И. [и др.] Система BASIC-6 как универсальный диалоговый процессор. // «Кибернетика», 1976, №3, с.29-32
36. Кетков Ю.Л., Рябов А.Н. Программирование на БЭЙСИКЕ: Входной язык системы программирования БЭЙСИК-222. Учебное пособие. – Горький, 1977. – 73 с
37. Кетков Ю.Л., Максимов В.С., Рябов А.Н. Входной язык программирования БЭЙСИК-222. // ГосФАП, рег. номер П003233, 1978
38. Кетков Ю.Л., Рябов А.Н. Эффективная система программирования BASIC - 222. //Программирование, 1978, №1.
39. Кетков Ю.Л. Программирование на БЭЙСИКЕ. –М.: Статистика, 1978. – 178 с
40. Ильин Э.Н., Кетков Ю.Л., Максимов В.С. Опыт разработки системы программирования БЭЙСИК-ЕС. – Всесоюзная конференция «Методы трансляции». Новосибирск, 1981
41. Ильин Э.Н., Кетков Ю.Л., Максимов В.С., Максимова Ю.В., Тафорина Н.М. Основные характеристики диалоговой системы программирования БЭЙСИК-ЕС. – 3-я Всесоюзная конференция «ДИАЛОГ-83». Протвино, 1983
42. Баяковский Ю.М., Галактионов В.А., Михайлова Т.Н. Графор. Графическое расширение Фортрана. – М.: Наука. Главная редакция физ.-мат. литературы, 1985. – 288 с.

43. Максимов В.С. Использование машинной графики в системах проектирования на базе ЕС ЭВМ. Учебное пособие. – Горький: Изд. ГГУ, 1979
44. Кетков Ю.Л. Программные средства для организации диалога с прикладной программой под управлением ДООС ЕС. – В сб «Оптимизация и математическое обеспечение САПР». – Горький: изд. ГГУ, 1980
45. Кетков Ю.Л. Пакет программ ДИАФОР для организации диалога под управлением ДООС ЕС. // ГосФАП, рег. номер П005892, 1983
46. Зверев В.И., Кетков Ю.Л., Максимов В.С. Алфавитно-цифровые дисплеи ЕС-7920 в диалоговых системах. – М.: Наука, 1986. – 240 с.
47. Кетков Ю.Л., Куракина И.М. Программирование на алгоритмических языках БЕЙСИК и ФОРТРАН. Учебное пособие. – Горький: изд. ГГУ, 1983. – 104 с
48. Кетков Ю.Л. Графические средства в алгоритмическом языке Бейсик. // Микропрограммные средства и системы, 1988, №1
49. Кетков Ю.Л. Три поколения систем программирования на базе языка Бейсик. – Всесоюзная конференция «Методы трансляции и конструирования программ». – Новосибирск, 1988
50. Кетков Ю.Л. Размышления о БЕЙСИКЕ (три поколения популярного языка программирования). – В сб. «Математическое моделирование и оптимизация». – Горький: изд. ГГУ, 1990
51. Кетков Ю.Л., Максимов В.С. Организация и работа с данными в специализированной САПР. – Всесоюзная конференция «Математическое обеспечение моделирования сложных систем», Киев, 1977
52. Ильин Э.Н., Кетков Ю.Л., Максимов В.С., Рябов А.Н. Архитектура системы коллективного пользования на ЕС ЭВМ. – Всесоюзная конференция «Математическое обеспечение моделирования сложных систем», Киев, 1977
53. Кетков Ю.Л. Диалог с прикладной программой под управлением ДООС ЕС. – Всесоюзная конференция «Методы мат. логики в проблемах искусственного интеллекта систематического программирования». Паланга, 1980
54. Ильин Э.Н., Кетков Ю.Л., Куркина Т.А., Максимов В.С., Тафорина Н.М. Расширение возможностей ФОРТРАНА ЕС ЭВМ для программирования диалоговых сценариев. – там же
55. Гордион М.М., Кетков Ю.Л., Куракина И.М., Молоткова Т.Д., Тафорина Н.М. Расширение возможностей ППП ГРАФОР для воспроизведения полигональных линий и надписей. – 1-я Всесоюзная конференция «Методы и средства обработки СГИ». Горький, 1983
56. Антипова Р.И., Белова М.Л., Гордион М.М., Ильин Э.Н., Кетков Ю.Л., Куракина И.М., Молоткова Т.Д., Тафорина Н.М. Инструментальная под-

- система для вывода картографической информации на ЕС ЭВМ. – там же
57. Антипова Р.И., Ильин Э.Н., Кетков Ю.Л., Куракина И.М. Унифицированная схема воспроизведения условных дискретных знаков топографических карт. – там же
 58. Кетков Ю.Л., Максимов В.С. Макроязык описания графических элементов. – 2-я Всесоюзная конференция «Методы и средства обработки СГИ». Горький, 1985
 59. Васин Ю.Г., Башкиров О.А., Кетков Ю.Л., Кобрин Р.Ю., Кустов Е.А., Ясаков Ю.В. Автоматизированная система обработки информации морских карт (АСОИМК). – 3-я Всесоюзная конференция «Методы и средства обработки СГИ». Горький, 1988
 60. Васин Ю.Г., Кетков Ю.Л., Куракина И.М. КАРТ-ДОК - автоматизированная система контроля и издания цифровых топографических карт. – 2-я Всероссийская конференция «Распознавание образов и анализ изображений: новые информационные технологии», Ульяновск, 1995 (труды конференции, ч.4).
 61. Антипова Р.И., Ильин Э.Н., Кетков Ю.Л. Оптимизация времени подготовки дисплейных кадров для визуального контроля картографических материалов в системе КАРТ-ДОК. – там же (труды конференции, ч.3)
 62. Васин Ю.Г., Кетков Ю.Л., Куракина И.М. Технология компьютерной подготовки издательских оригиналов по цифровым моделям карт местности. – там же
 63. Васин Ю.Г., Кетков Ю.Л., Куракина И.М. КАРТ-ДОК: система визуального контроля, обновления и подготовки к изданию цифровых топографических карт и планов городов. – Всероссийская конференция КОГРАФ-98. – Н.Новгород, 1998
 64. Кетков Ю.Л., Кирьянов С.К. Оптимизация времени отображения векторных графических изображений большого формата. Труды 11-й международной конференции по компьютерной графике и машинному зрению ГРАФИКОН'2001, Н. Новгород, 2001
 65. Y. Ketkov, M. Tyulin. Real-time visualization of large vector HP-GL based maps. «Труды 15-й Международной конференции по компьютерной графике и ее приложениям GraphiCon'2005», Новосибирск, июнь 2005 г., Институт вычислительной математики и математической геофизики, 2005. – pp. 316-317
 66. Кетков Ю.Л., Кузнецов А.И. Создание графических примитивов в формате HP-GL. Там же, – с. 378-381

67. Кетков Ю.Л., Максимов В.С. Организация и работа с данными в специализированной САПР. – В сб. «Оптимизация и математическое обеспечение САПР». – Горький: изд. ГГУ, 1978
68. Антипова Р.И., Белова М.Л., Гордион М.М., Ильин Э.Н., Кетков Ю.Л., Куракина И.М., Молоткова Т.Д., Тафорина Н.М. Инструментальная подсистема для вывода картографической информации на ЕС ЭВМ. – В сб. «Автоматизация обработки сложной графической информации». – Горький: изд. ГГУ, 1987
69. Кетков Ю.Л. Создание инструментальных программных средств для разработки диалоговых систем САПР и АСНИ. – Диссертация на соискание ученой степени доктора технических наук в форме научного доклада, Н.Новгород, 1992
70. Кетков Ю.Л. Обработка временных интервалов и календарных дат на IBM PC. // Начертательная геометрия, инженерная и компьютерная графика. Международный межвузовский сб. трудов кафедр графических дисциплин, Н. Новгород, ННГАСУ, вып. 4, 2000. – с. 107-119 (Миф 2000)
71. Кетков Ю.Л., Матвеев З.А. Методы повышения эффективности систем воспроизведения картографических документов. // Вестник ННГУ. - изд. ННГУ, № 2, 2008, с. 138-146
72. Задыхайло И.Б., Камынин С.С., Кетков Ю.Л., Корягин Д.А., Любимский Э.З., Шура-Бура М.Р. Математическое обеспечение. Операционная система БЭСМ-6 Монитор. Общее описание. – М.: Препринт ИПМ АН СССР, 1968. – 28 с.
73. Кетков Ю.Л., Лебедев К.В., Петрова Е.В., Уварова Е.П. Технология визуального программирования в среде Delphi. Учебное пособие. – Н.Новгород: изд. ННГУ, 1997. – 138 с
74. Кетков Ю.Л., Максимов В.С., Рябов А.Н. Введение в системное программирование на языке АССЕМБЛЕРА ЕС ЭВМ. – М.: Наука, 1982. – 264 с.
75. Кетков Ю.Л. Диалог на языке Бейсик для мини- и микро-ЭВМ. – М.: Наука, 1988. – 368 с.
76. Кетков Ю.Л. GW-, Turbo- и Quick-BASIC на IBM PC/XT и AT. – М.: Финансы и статистика, 1992. – 240 с.
77. Кетков Ю.Л. Толковый словарь языка программирования БЕЙСИК. – М.: Наука, 1992. – 320 с.
78. Кетков Ю.Л., Кетков А.Ю., Шапошников Д.Е. Школьная энциклопедия: Персональный компьютер. – М.: Большая Российская Энциклопедия, изд. Дом Дрофа, Москва, 1997, 1998. – 440 с.

79. Кетков Ю.Л., Кетков А.Ю. Практика программирования: Бейсик, Си, Паскаль. Самоучитель. – Санкт-Петербург: БХВ-Петербург, 2001, 2002. – 480 с
80. Кетков Ю.Л., Кетков А.Ю. Практика программирования: Visual Basic, С++ Builder, Delphi. Самоучитель. – Санкт-Петербург: БХВ-Петербург, 2002. – 464 с.
81. Кетков Ю.Л., Кетков А.Ю., Шульц М.М. MatLab 6х: программирование численных методов. – Санкт-Петербург: БХВ-Петербург, 2004. – 672 с.
82. Кетков Ю.Л., Кетков А.Ю., Шульц М.М. MatLab 7: программирование, численные методы. – Санкт-Петербург: БХВ-Петербург, 2005. – 752 с.
83. Кетков Ю.Л. Введение в языки программирования С и С++. Интернет-университет информационных технологий - ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2006 г., 344 стр.
84. Кетков Ю.Л., Кетков А.Ю. Свободное программное обеспечение: Free Pascal для студентов и школьников. – Санкт-Петербург: БХВ-Петербург, 2010. – 384 с.

ПУБЛИКАЦИИ Ю.Л. КЕТКОВА

*Включены отдельные
публикации, начиная
с самых первых (1958 г.),
а также неопубликованные
доклады и заметки*

Об одном этюде для программистов

УДК 681.3.06

ОБ ОДНОМ ЭТЮДЕ ДЛЯ ПРОГРАММИСТОВ

Кетков Ю.Л.

Среди многочисленной литературы по математическому обеспечению ЭВМ не так уж часто встречаются работы, привлекающие нас такими названиями как "Искусство программирования для ЭВМ" (Д. Кнут), "Программирование - вторая грамотность" (А.П. Ершов), "Этюды для программистов" (Ч. Уэзерелл). Встреча с ними тем более приятна, что за интересными названиями скрываются и поистине энциклопедический материал, сопровождаемый разбором различных вариантов тщательно подобранных задач, и анализ эффективности получающихся программ, и демонстрация полезных технологических приемов программирования, и неожиданные параллели, способствующие углубленному восприятию предмета.

К одной из наиболее удачных новинок такого плана относится вышедшая недавно в издательстве "Мир" книга Ч.Уэзерелла^[1]. Как только она появилась в продаже, у многих программистов возникло желание "расправиться" с наиболее трудным, на наш взгляд, этюдом № 9 - "Познай самого себя". Его формулировка укладывается в одну строку - составить программу, печатающую свой собственный текст. Подзадоривало и отсутствие в книге решения, и предложение не отчаиваться после 13 неудачных попыток, и желание улучшить предлагаемый Уэзереллом объем программы в 30-40 операторов. Поэтому на следующий день после покупки книги среди сотрудников отдела математического обеспечения был объявлен конкурс.

Первое решение, принадлежащее автору настоящей заметки и оказавшееся далеко не лучшим, появилось через день. Для его реализации

- 2 -

был использован ФОРТРАН ЭС ЭВМ. Было ясно, что с помощью основной пары операторов *PRINT* и *FORMAT* в рамках одной программы задачу решить трудно, т.к. при этом возникали бесконечные вложения типа:

```
1 FORMAT(... PRINT 1,... 1 FORMAT(... PRINT 1 ,...
```

Не выручала и замена холлеритовых констант текстом, заключенным в кавычки, т.к. это приводило к бесконечно повторяющимся кавычкам.

Поэтому основная идея первого решения сводилась к тому, чтобы разбить программу на две части - головную программу и подпрограмму печати *PRINT*. Головная программа состояла из однотипных операторов обращения к подпрограмме печати:

```
CALL PRINT (n1, n1 Hтекст1)
CALL PRINT (n2, n2 Hтекст2)
. . . . .
CALL PRINT (nk, nk Hтекстk)
END
```

В качестве параметров в обращении присутствуют тексты операторов подпрограммы *PRINT* (текст_i) и количество символов *n_i* в передаваемом тексте.

Задача подпрограммы печати заключалась в воспроизведении переданного ей текста с добавкой стандартной приставки *CALL PRINT* и круглых скобок, размещающихся в нужных позициях выводной строки. Помимо этого очередной оператор, переданный в подпрограмму печати, накапливался в специально выделенном массиве *A*. Получив от вызывающей программы текст последнего оператора (*ZHEND*), подпрограмма выдавала его на печать уже без обрамления и распечатывала свой собственный текст, накопившийся в массиве *A*. Для работы с текстовыми данными пришлось использовать однобайтовые логические переменные. Технология сравнения логических переменных на ФОРТРАНЕ достаточно подробно описана в работе [2].

Полученное таким образом решение казалось естественным и даже красивым (по мнению автора), т.к. удалось обойтись без рекурсии, на которую намекал Уэзерелл. Однако это решение не удовлетворяло условию длины, т.к. содержало 45 операторов. Тем не менее предложенный вариант был обнародован и послужил материалом для размышлений. По его следам для ЭВМ М7000 была составлена разноязычная программа Ю.Г. Гервагиним. Подпрограмма печати передаваемого ей текстового массива была написана на ФОРТРАНЕ (9 операторов), а головная программа, содержащая единственный оператор вызова и набор текстовых констант, была составлена на МНЕМОКОДЕ и содержала 23 оператора. В этом варианте было два нюанса. Во-первых, подпрограмма вызывалась один раз и после распечатки передаваемого ей массива прекращала работу по оператору `STOP`. Во-вторых, текстовый массив состоял из строк равной длины (по 24 символа).

Идею с подпрограммой можно было совершенствовать. Например, написать вызывающую программу также на ФОРТРАНЕ и вместо многочисленных операторов `CALL`, передающих на печать по одной строке, воспользоваться одним оператором `CALL` с продолжением. Общее число операторов при этом, повидимому, удалось бы свести к 10-15. По такому принципу В.П. Новиним была построена программа, которая рисует свой собственный исходный текст на графопостроителе с помощью ППП ГРАФОР^[3].

Графический вариант состоит из двух программных единиц. Подпрограмма типа `BLOCK DATA` присваивает однобайтовым логическим переменным (одномерные массивы `A`, `B`, `C`, ...) значения, соответствующие текстам операторов головной программы. Содержит подпрограмма всего 4 оператора, хотя один из них (`LOGICAL`) имеет 13 строк продолжения. Головная программа извлекает из блока общей памяти текст своего очередного оператора и с помощью подпрограммы `SYMBOL`

воспроизводит этот текст в верхней и нижней части страницы. В одном случае этот текст воспроизводится без каких-либо добавлений, и это соответствует тексту оператора головной программы. В другом случае текст обрамляется слева и справа стандартно формируемыми добавлениями. Это соответствует употреблению текста в подпрограмме. Обратим внимание на некоторые нюансы графического варианта. Во-первых, в связи с тем, что длина текстовой константы в ФОРТРАНЕ ЕС не может превышать 256 байтов, пришлось текст программы раздробить по строкам и ввести для этой цели одномерные массивы А, В, Во-вторых, чтобы не усложнять в головной программе работу с большим количеством введенных массивов, пришлось поставить им в соответствие объединенный массив Т. Просто с помощью оператора EQUIVALENCE этого достичь не удастся, если только не отождествлять соответствующие элементы массива Т с первыми элементами каждого из массивов А, В, С, В противном случае ($T \equiv A$) транслятор распределяет память под элементы массивов В, С, ... вслед за последним элементом массива Т, т.е. строки с текстом исходной программы оказываются разорванными. Поэтому и пришлось прибегнуть к услугам общей памяти. Наконец, третий нюанс связан с приформированием к тексту оператора стандартных окаймлений слева и справа. Здесь помогла возможность задавать не абсолютные координаты начала текста, а признак размещения очередной части строки, начиная с текущего положения пишущего механизма. Такой режим в подпрограмме SYMBOL предусмотрен, и он облегчил проблему вывода составных строк.

После этих вариантов, исчерпавших идеи с подпрограммой и удовлетворявших условиям длины по Уззереллу, страсти немного улеглись. Вскоре мне довелось встретиться с редактором перевода Д.М. Баяковским, который получил поступившее в редакцию письмо из Караганды. Его автор - Е.В. Пешехонов, предложил казалось бы нелучша-

емый вариант, реализованный на ПЛ/И и содержащий всего 5 операторов. В печатном документе текст программы занимает всего 5 строк, но из соображений наглядности мы приведем его в несколько более структурированном формате:

```
A: PROC _ OPTIONS (MAIN);
DCL _ C _ CHAR (166) INIT ('
  A: PROC _ OPTIONS (MAIN);
  DCL _ C _ CHAR (166) INIT ('');
  OPEN _ FILE (SYSPRINT) LINESIZE (71);
  PUT _ EDIT (SUBSTR (C, 1, 42),
              SUBSTR (C, 1, 42),
              SUBSTR (C, 42, 125),
              SUBSTR (C, 42, 125)) (PAGE, 4A);
  END _ A; _ ');
OPEN _ FILE (SYSPRINT) LINESIZE (71);
PUT _ EDIT (SUBSTR (C, 1, 42),
            SUBSTR (C, 1, 42),
            SUBSTR (C, 42, 125),
            SUBSTR (C, 42, 125)) (PAGE, 4A);
END _ A;
```

Как видно из приведенного текста, программа включает константу, почти полностью повторяющую текст программы. В этой константе в соответствии с синтаксисом языка для воспроизведения кавычки приходится употребить двойную кавычку. Однако в рабочей программе двойная кавычка будет заменена одним символом. Поэтому на печать кавычку приходится выдавать дважды - один раз как последний элемент в подстроке `SUBSTR (C, 1, 42)`, а второй раз как первый элемент в подстроке `SUBSTR (C, 42, 125)`.

Вариант Е.В.Пешехонова вызвал вторую волну энтузиазма, охватившую не только научных сотрудников, но и студентов Горьковского университета. Совершенно независимо от варианта Е.В.Пешехонова была составлена программа также на ПЛ/И студентом 3 курса Н.В.Маковеевым,

- 6 -

занявшим 3-е место в конкурсе. Вот ее текст:

```
P:PROC_OPTIONS(MAIN); DCL T(6) CHAR(60), I S, I2 # CHAR(10)
  I2 B CHAR(2), I2 C CHAR(3), I2 A CHAR(1), I2 D CHAR(2);
T(I) = ' P:PROC_OPTIONS(MAIN); ... } текст первых двух ;
T(I2) = ' I2 B CHAR(2), I2 C ... } строк программы ;
T(I3) = ' DO I=1 TO I2; ... } текст последних четырех ;
T(I4) = ' PUT EDIT(B,I, ... } строк программы ;
T(I5) = ' PUT EDIT(#,A,#,... ;
T(I6) = ' DO I=3 TO I6; ... ;
STRING(S) = 'STRING(S)=T( ) = ''''''';
DO I=1 TO I2, PUT EDIT(T(I)) (SKIP,A); END; DO I=1 TO I6;
PUT EDIT(B,I,C,T(I),D) (SKIP,X(1),A,F(2),I3A); END;
PUT EDIT(#,A,#,B,C,A,A,A,D,D) (SKIP,X(1),I4A);
DO I=3 TO I6, PUT EDIT(T(I)) (SKIP,A); END; END P;
```

Интересной особенностью этой программы является привлечение механизма структур. Структура S состоит из пяти элементов с именами #, B, C, A и D соответственно. После выполнения оператора `STRING(S) = ...` эти элементы принимают следующие значения:

# :	STRING(S) =	(10 символов)
B :	T((2 символа)
C :) ='	(3 символа)
A :	'	(1 символ)
D :	';	(2 символа)

В дальнейшем элементы B, C и счетчик цикла I используются для формирования левой части в операторах присваивания (строки 3-8). Так как значение I выдается на печать по формату F(2), то именно этим объясняется наличие лишнего пробела в операторах присваивания `-T(I)`. В эту позицию при выводе помещается знак I. Для печати оператора `STRING(S) = ...` используются все элементы структуры S. Остальные операторы печати (строки I0 и I7) в комментариях

- 7 -

не нуждаются.

И все-таки, несмотря на универсальность языка ПЛ/I, пальма первенства осталась за ФОРТРАНОМ. На этом языке В.С.Максимову удалось написать программу, содержащую всего 3 (!) оператора:

```
LOGICAL=1P(112)/112H(7X,66A1/6X,1H*,66A1/6X,2H*/17X,39A1/7X,3HEND
*10A1)...LOGICAL=1P(112)/112HPRINTP(P(I),I=54,73),P,(P(I),I=74,113)
*/
PRINTP,(P(I),I=54,73),P,(P(I),I=74,113)
END
```

Не будем осуждать победителя за то, что из-за отсутствия пробелов текст программы оказался не очень удобочитаем. Однако согласитесь, что 3 оператора на ФОРТРАНЕ - это абсолютный рекорд, более чем в 10 раз превышающий установку Уэзерелла.

Из-за отсутствия под рукой журнала, в котором приводится решение этюда, принадлежащее П.Брэтли и Дж.Нило^[4], нам не удалось сравнить отечественные результаты с зарубежными. Однако я несколько не сомневался в победе нашей "команды".

Публикация настоящей заметки преследует две цели. Во-первых, хотелось привлечь внимание широких программистских кругов к различным вариантам решения довольно интересной задачи и побудить их к разработке оригинальных этюдов. Это будет полезно и для начинающих и для опытных программистов, которым не мешало бы перенять богатейший опыт наших шахматистов. Во-вторых, хотелось бы лишний раз подчеркнуть, что при надлежащей постановке разработка программного обеспечения в нашей стране может и должна находиться на уровне лучших мировых стандартов.

Литература.

1. Уэзерелл Ч. Этюды для программистов: Пер. с англ. под ред. Ю.М.Баяковского - М.: Мир, 1982.

- 8 -

2. Кетков Ю.Л., Максимов В.С., Рябов А.Н. Введение в системное программирование на языке ассемблера ЕС ЭВМ. - М.: Наука. Главная редакция физ.-мат. литературы, 1982.
3. Баяковский Ю.М., Михайлова Т.Н., Мишакова С.Т. ГРАФОР: комплекс графических программ на ФОРТРАНЕ. Вып. I. Основные элементы графики. Препринт № 41, ИПМ АН СССР, М., 1972.
4. Bratley P., Mello J. Computer Recreations Self-Reproducing Automata. Software - Practice and Experience, 2, pp. 397-400, 1972

Об одном способе вычисления полиномов на математических машинах

ИЗВЕСТИЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ

Том I, № 4

РАДИОФИЗИКА

1958

ОБ ОДНОМ СПОСОБЕ ВЫЧИСЛЕНИЯ ПОЛИНОМОВ НА МАТЕМАТИЧЕСКИХ МАШИНАХ

Ю. Л. Кетков

Приводится описание нескольких методов вычисления значений полиномов, позволяющих получить значительную экономию времени при производстве расчетов на современных вычислительных машинах.

При численном решении математических задач зачастую приходится сталкиваться с вычислением полиномов, иногда довольно высоких степеней. К этому приводит вычисление функций, заданных конечным отрезком ряда Тэйлора или аппроксимированных различными интерполяционными или специальными полиномами. Такого рода вычисления обычно приходится проводить для многих значений аргумента.

Для того, чтобы вычислить значение полинома n -ой степени

$$f_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0,$$

обычно пользуются рекуррентными соотношениями

$$f_0 = a_n x;$$
$$f_{k+1} = f_k x + a_{n-k-1} \quad (k = 0, 1, \dots, n-1).$$

Этими соотношениями пользовался еще Ньютон, хотя в математической литературе они известны как правило Горнера [1].

Простота программирования и сравнительно небольшое число операций, необходимых для реализации вычисления, привели к широкому распространению правила Горнера при вычислениях на современных электронных цифровых машинах. В трехадресных вычислительных машинах программа, использующая это правило, занимает всего 3–4 приказа (длина программы не зависит от степени полинома).

При использовании правила Горнера вычисление $f_n(x)$ занимает ровно n умножений и n сложений. При $a_n = 1$ число умножений уменьшается до $n-1$. В дальнейшем, не ограничивая общности, будем рассматривать полиномы, у которых $a_n = 1$.

Большой практический интерес представляет задача улучшения правила Горнера, т. е. нахождения правила, использующего только умножения и сложения, при котором потребуется меньше, чем n , умножений, или меньше, чем n , сложений, или, по крайней мере, доказательства существования такого правила. При этом гораздо выгоднее идти по линии уменьшения числа умножений, так как эта операция является более трудоемкой, нежели сложение. В последовательных вычислительных машинах умножение занимает в m раз больше времени, чем сложение (m — наибольшее число разрядов, допускаемых в представлении чисел в машине). В параллельных вычислительных машинах это соотношение меньше (порядка 3–5).

1. В [1] делается попытка доказать, что правило Горнера улучшить (в указанном выше смысле) невозможно. Для любого n

доказывается, что нельзя построить правило вычисления полиномов, использующее меньше, чем n сложений. Относительно умножения такая задача только ставится; выдвинутое предположение доказано лишь для $n > 4$.

Однако, не противореча указанным результатам, постановку задачи можно представить несколько иначе. Вычисление полинома можно провести в два этапа. На первом этапе производится предварительная обработка коэффициентов исходного полинома; для этого указываются формулы пересчета исходных коэффициентов a_0, a_1, \dots, a_{n-1} в некоторые другие $-b_0, b_1, \dots, b_k$. На втором этапе производится собственно вычисление полинома с использованием уже введенных коэффициентов. Если при этом будет использовано меньше, чем $n-1$ умножение, то имеет смысл произвести предварительную обработку (может быть и довольно сложную) с тем, чтобы в дальнейшем можно было получить значительную экономию времени работы машины. Экономия будет тем больше, чем в большем количестве точек необходимо вычислить данный полином.

2. Обозначим через λ_n число умножений, используемых для вычисления полинома n -ой степени, и попытаемся оценить минимальное значение λ_n . Собственно к этому и сводится первая часть задачи „улучшения“ правила Горнера. Вторая и более ценная часть заключается в построении метода, использующего это минимальное число умножений.

Из сравнительно простых рассуждений можно получить следующие оценки для λ_n .

а) Непосредственно из правила Горнера следует, что

$$\min \lambda_n \leq n - 1.$$

б) Обозначим через Ψ_n минимальное число умножений, необходимых для получения x^n из x с помощью только умножений. Нетрудно убедиться, что при этом

$$[\log_2 n] \leq \Psi_n \leq 2[\log_2 n],$$

где нижняя оценка достигается при $n=2^p$ (p —целое положительное число), а верхняя — при $n=2^p - 1$. Ввиду того, что $f(x) = x^n$ есть полином n -ой степени,

$$\min \lambda_n \geq \Psi_n.$$

в) Менее очевидной является оценка

$$\min \lambda_n \geq [(n+1)/2].$$

В общем случае способ вычисления полиномов, использующий только сложения и умножения, можно представить следующим образом. Исходными величинами являются x, b_0, b_1, \dots, b_k . Вычисление полинома складывается из последовательных шагов, на каждом из которых производится следующие операции: берутся некоторые два полинома из совокупности всех, вычисленных на предыдущих шагах; к каждому из них можно прибавить по одному* из коэффициентов $b_j (j=0, 1, \dots, k)$ и сколько угодно других полиномов, вычисленных ранее; полученные суммы перемножаются; к произведению можно прибавить произвольное число уже вычисленных выражений; результат присоединяется к уже вычисленным полиномам, после чего переходят к следующему шагу.

* Заметим, что в процессе вычисления не имеет смысла прибавлять на некотором шаге комбинацию коэффициентов $b_{j_1} + b_{j_2} + \dots + b_{j_m}$ и больше нигде (ни в предыдущих, ни в последующих шагах) ни один из этих коэффициентов не использовать. Простым переобозначением всегда можно свести дело к прибавлению только одного коэффициента (при условии, конечно, что исходные данные не связаны никакой функциональной зависимостью).

В результате на каком-то шаге требуемый полином окажется вычисленным. Его коэффициенты получаются при таком вычислении в виде некоторых функций (многочленов) от b_j . Эти функции не должны быть связаны между собой никакой зависимостью, так как коэффициенты полинома есть, вообще говоря, произвольный набор из n чисел. Отсюда вытекает, что число вводимых коэффициентов b_j должно быть не менее n . В предложенной выше схеме вычисления каждый шаг позволяет ввести не более двух коэффициентов b_j . Таким образом, если считать, что предложенная схема является самой общей среди всех схем, использующих операции сложения и умножения, то требуется сделать минимум $[(n+1)/2]$ шагов для введения не менее n коэффициентов b_j , т. е. затратить не менее $[(n+1)/2]$ умножений.

3. Покажем на конкретных примерах, как иногда удается вычислить значение полинома с использованием $[(n+1)/2]$ умножений.

Первый пример ($n=4$):

$$P_4 = x^4 + a_3x^3 + a_2x^2 + a_1x + a_0.$$

Схема вычисления P_4 такова:

$$N_2 = x(b_0 + x); \quad N_4 = (N_2 + x + b_1)(N_2 + b_2) + b_3.$$

Формулы для пересчета коэффициентов получаются из условия $N_4 = P_4$. Приравнявая коэффициенты при одинаковых степенях x , получаем

$$(4.1) \quad a_3 = 2b_0 + 1;$$

$$(4.2) \quad a_2 = b_1 + b_2 + b_0(b_0 + 1);$$

$$(4.3) \quad a_1 = b_0(b_1 + b_2) + b_3;$$

$$(4.4) \quad a_0 = b_1b_2 + b_3.$$

Уравнения очень просто разрешаются относительно b_0, b_1, b_2, b_3 и если исходные коэффициенты a_0, a_1, a_2, a_3 были действительными, то и коэффициенты b_0, b_1, b_2, b_3 также получаются действительными. Введение слагаемого x при вычислении N_4 делает совместными уравнения (4.2) и (4.3).

Приведенная схема позволяет вычислить (после предварительной обработки) любой полином четвертой степени за два умножения и пять сложений, тогда как схема Горнера использует три умножения и четыре сложения.

Второй пример ($n=5$):

$$P_5 = x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0.$$

Схема вычисления P_5 следует из схемы вычисления P_4 , если учесть, что $P_5 = x\tilde{P}_4 + a_0$, где

$$\tilde{P}_4 = x^4 + a_4x^3 + a_3x^2 + a_2x + a_1.$$

При этом для вычисления P_5 необходимо затратить (после предварительной обработки) три умножения и шесть сложений, тогда как схема Горнера потребовала бы использования четырех умножений и пяти сложений.

Третий пример ($n=6$):

$$P_6 = x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0.$$

Схема вычисления P_6 :

$$N_2 = x(b_0 + x);$$

$$N_4 = (N_2 + x + b_1)(N_2 + b_2) + b_3;$$

$$N_6 = (N_2 + b_4)N_4 + b_5.$$

Соотношения между коэффициентами:

$$(6.1) \quad a_5 = 3b_0 + 1;$$

$$(6.2) \quad a_4 = b_1 + b_2 + b_4 + 2b_0 + 3b_0^2;$$

$$(6.3) \quad a_3 = b_2 + b_4 + 2b_0(b_1 + b_2 + b_4) + b_0^2 + b_0^3;$$

$$(6.4) \quad a_2 = b_2b_4 + b_3 + (b_0 + b_1)(b_2 + b_4);$$

$$(6.5) \quad a_1 = b_2b_4 + b_3b_1(b_2 + b_4) + b_0(b_2b_4 + b_3);$$

$$(6.6) \quad a_0 = b_1b_2b_4 + b_3b_4 + b_3.$$

Уравнения легко разрешаются методом подстановок; однако при нахождении b_2 и b_4 приходится решать квадратное уравнение, которое может и не иметь действительных корней. Для того, чтобы b_2 и b_4 получились действительными (при условии, что исходные коэффициенты действительны), надо потребовать, чтобы дискриминант этого квадратного уравнения не был отрицателен. Это требование налагает на коэффициенты исходного полинома связь типа $\varphi(a_1, a_2, a_3, a_4, a_5) > 0$. Геометрически этот факт можно истолковать следующим образом. Каждому полиному P_6 поставим в соответствии точку (a_0, a_1, \dots, a_5) шестимерного пространства. Тогда для полиномов, „лежащих“ по одну сторону от поверхности $\varphi(a_1, a_2, \dots, a_5) = 0$ и на ней самой, применение указанной выше схемы позволяет пересчитать действительные исходные коэффициенты также в действительные. Для полиномов, „лежащих“ по другую сторону от этой поверхности, пересчет приводит к комплексным выражениям для b_2 и b_4 .

В области применимости* схема позволяет вычислять полиномы шестой степени с помощью трех умножений и семи сложений, тогда как схема Горнера требует пяти умножений и шести сложений.

Четвертый пример ($n=7$):

Схема вычисления P_7 вытекает из схемы вычисления P_6 , если заметить, что $P_7 = x\tilde{P}_6 + a_0$, где

$$\tilde{P}_6 = x^6 + a_6x^5 + a_5x^4 + a_4x^3 + a_3x^2 + a_2x + a_1.$$

Замечания относительно области применимости схемы вычисления P_6 полностью переносятся на схему вычисления P_7 .

Пятый пример ($n=8$):

$$P_8 = x^8 + a_7x^7 + \dots + a_1x + a_0.$$

Схема вычисления P_8 :

$$N_2 = x(b_0 + x);$$

$$N_4 = (N_2 + x + b_1)(N_2 + b_2) + b_3;$$

$$N_6 = (N_2 + b_4)N_4 + b_5;$$

$$N_8 = (N_2 + b_6)N_6 + b_7.$$

Ввиду громоздкости не будем приводить соотношения между коэффициентами, но заметим, что при пересчете приходится решать квадратное уравнение. Таким образом, имеет место замечание, аналогичное замечанию в третьем примере с $n=6$.

* Под областью применимости понимается та часть пространства коэффициентов, для которой пересчет не выводит из поля действительных чисел.

В области применимости приведенная схема позволяет вычислять полиномы восьмой степени с помощью четырех умножений и девяти сложений.

Шестой пример ($n=9$):

Схема вычисления P_9 строится из схемы вычисления P_8 так же, как и в примерах 2 и 4.

Несмотря на значительную экономию при многократных вычислениях, приведенные выше способы (для $n \geq 6$) не всегда могут быть применены. Их использование становится невыгодным, если при пересчете будут получены комплексные выражения для некоторых коэффициентов b_j .

Однако нетрудно указать способ (более экономный, нежели схема Горнера) пересчета коэффициентов, который позволяет из действительных исходных коэффициентов получать также действительные. Он непосредственно вытекает из возможности представления любого полинома P_n в виде:

$$P_n = P_3 \prod_{k=1}^l P_4^{(k)}, \quad (1)$$

где P_3 — полином степени не выше третьей, $P_4^{(k)}$ — полином четвертой степени, $l = [n/4]$.

Для вычисления выражения, стоящего в правой части, необходимо произвести λ_n умножений, причем

$$3[n/4] < \lambda_n < 3[n/4] + 2,$$

и ν_n сложений, причем

$$5[n/4] < \nu_n < 5[n/4] + 3.$$

Верхние грани в обеих оценках достигаются при $\nu = 3$, нижние — при $\nu = 0$.

Применение этого способа удобно, если известны корни P_n . При этом упрощается сведение P_n к виду (1).

4. В случае, если корни P_n неизвестны и сведение к виду (1) затруднительно, можно предложить следующую схему вычисления.

Пусть $n = 2k$ и

$$P_n = x^{2k} + a_{2k-1}x^{2k-1} + \dots + a_0.$$

Схема вычисления P_n такова:

$$N_2 = x(b_0 + x);$$

$$N_4 = (N_2 + b_1 + x)(N_2 + b_2) + b_2;$$

$$N_6 = N_2 N_4 + b_3 x + b_3;$$

$$N_8 = (N_2 + b_4) N_6 + b_4;$$

$$N_{10} = N_2 N_8 + b_5 x + b_5;$$

$$\dots$$

$$N_{2k} = (N_2 + \delta_k b_{2k-2}) N_{2k-2} + \bar{\delta}_k b_{2k-2} x + b_{2k-1},$$

где $\delta_k = 0$, $\bar{\delta}_k = 1$, если k — четное, и $\delta_k = 1$, $\bar{\delta}_k = 0$, если k — нечетное. Из условия $P_n = N_{2k}$ находятся соотношения между введенными и исходными коэффициентами.

Обозначим через $b_0^{(l)}$, $b_1^{(l)}$, ..., $b_{2l-1}^{(l)}$ коэффициенты промежуточного полинома N_{2l} ($l=1, 2, \dots, k$) и предположим для определенности, что k — нечетное. Пересчет коэффициентов исходного полинома

предлагается вести следующим образом. Рассмотрим соотношения между коэффициентами N_{2k} и N_{2k-2} . Из

$$N_{2k} = (x^2 + b_0 x) N_{2k-2} + b_{2k-2} x + b_{2k-1}$$

имеем:

$$\left. \begin{aligned} (2k. 1) \quad a_{2k-1} &= b_{2k-1}^{(k)} = b_0 + b_{2k-3}^{(k-1)}; \\ (2k. 2) \quad a_{2k-2} &= b_{2k-2}^{(k)} = b_0 b_{2k-3}^{(k-1)} + b_{2k-4}^{(k-1)}; \\ (2k. 3) \quad a_{2k-3} &= b_{2k-3}^{(k)} = b_0 b_{2k-4}^{(k-1)} + b_{2k-5}^{(k-1)}; \\ &\dots \dots \dots \\ (2k. 2k-3) \quad a_3 &= b_3^{(k)} = b_0 b_2^{(k-1)} + b_1^{(k-1)}; \\ (2k. 2k-2) \quad a_2 &= b_2^{(k)} = b_0 b_1^{(k-1)} + b_0^{(k-1)}; \\ (2k. 2k-1) \quad a_1 &= b_1^{(k)} = b_0 b_0^{(k-1)} + b_{2k-2}; \\ (2k. 2k) \quad a_0 &= b_0^{(k)} = b_{2k-1}. \end{aligned} \right\} (A)$$

Проведя аналогичное рассмотрение только для предстарших коэффициентов всех промежуточных полиномов, приходим к

$$\left. \begin{aligned} b_{2k-1}^{(k)} &= b_0 + b_{2k-3}^{(k-1)}; \\ b_{2k-3}^{(k-1)} &= b_0 + b_{2k-5}^{(k-2)}; \\ &\dots \dots \dots \\ b_3^{(3)} &= b_0 + b_3^{(2)}; \\ b_0^{(2)} &= 2b_0 + 1. \end{aligned} \right\}$$

Отсюда $b_{2k-1}^{(k)} = kb_0 + 1$ и $b_0 = \frac{b_{2k-1}^{(k)} - 1}{k} = \frac{a_{2k-1} - 1}{k}$. Перейдя

к рассмотрению системы (A), из уравнения (2к. 1) находим $b_{2k-3}^{(k-1)}$. Подставляя уже известные величины в (2к. 2), получаем $b_{2k-4}^{(k-1)}$. Продолжая такие подстановки, из (2к. 2к-2) находим $b_0^{(k-1)}$. Из уравнения (2к. 2к-1) $-b_{2k-2}$ и из (2к. 2к) $-b_{2k-1}$. При всех этих вычислениях приходится решать только линейные уравнения. Таким образом, рассмотрение перехода от N_{2k} к N_{2k-2} позволяет вычислить $b_{2k-1}, b_{2k-2}, b_{2k-3}^{(k-1)}, b_{2k-4}^{(k-1)}, \dots, b_0^{(k-1)}$.

Рассмотрение перехода от N_{2k-2} к N_{2k-4} приводит к системе уравнений, несколько отличной от системы (A):

$$\left. \begin{aligned} (2k-2.1) \quad b_{2k-3}^{(k-1)} &= b_0 + b_{2k-5}^{(k-2)}; \\ (2k-2.2) \quad b_{2k-4}^{(k-1)} &= b_{2k-4} + b_0 b_{2k-5}^{(k-2)} + b_{2k-6}^{(k-2)}; \\ (2k-2.3) \quad b_{2k-5}^{(k-1)} &= b_{2k-4} b_{2k-5}^{(k-2)} + b_0 b_{2k-6}^{(k-2)} + b_{2k-7}^{(k-2)}; \\ (2k-2.4) \quad b_{2k-6}^{(k-1)} &= b_{2k-4} b_{2k-6}^{(k-2)} + b_0 b_{2k-7}^{(k-2)} + b_{2k-8}^{(k-2)}; \\ &\dots \dots \dots \\ (2k-2.2k-5) \quad b_3^{(k-1)} &= b_{2k-4} b_3^{(k-2)} + b_0 b_2^{(k-2)} + b_1^{(k-2)}; \\ (2k-2.2k-4) \quad b_2^{(k-1)} &= b_{2k-4} b_2^{(k-2)} + b_0 b_1^{(k-2)} + b_0^{(k-2)}; \\ (2k-2.2k-3) \quad b_1^{(k-1)} &= b_{2k-4} b_1^{(k-2)} + b_0 b_0^{(k-2)}; \\ (2k-2.2k-3) \quad b_0^{(k-1)} &= b_{2k-4} b_0^{(k-2)} + b_{2k-3}. \end{aligned} \right\} (B)$$

Решение системы (В) ведется также методом подстановок. Из уравнения $(2k-2.1)$ находим $b_{2k-5}^{(k-2)}$. Из уравнения $(2k-2.2)$ выражаем $b_{2k-6}^{(k-2)}$ как функцию b_{2k-4} . Подставляя найденные величины в $(2k-2.3)$, получаем $b_{2k-7}^{(k-2)}$ как функцию b_{2k-4} . Продолжая такие подстановки, из $(2k-2.2k-5)$ выражаем $b_1^{(k-2)}$, а из $(2k-2.2k-4) - b_0^{(k-2)}$ как функции относительно b_{2k-4} . Заметим, что все эти функции суть полиномы, а два последних имеют соответственно степени $k-3$ и $k-2$ относительно b_{2k-4} . Подстановка $b_1^{(k-2)}$ и $b_0^{(k-2)}$ в $(2k-2.2k-3)$ приводит к алгебраическому уравнению степени $k-2$ относительно b_{2k-4} . Поскольку k было выбрано нечетным, это уравнение имеет хотя бы один действительный корень, который можно взять в качестве b_{2k-4} . После того, как найдено b_{2k-4} , можно вычислить $b_{2k-6}^{(k-2)}, b_{2k-7}^{(k-2)}, \dots, b_1^{(k-2)}, b_0^{(k-2)}$ и, наконец, из уравнения $(2k-2.2k-2) - b_{2k-3}$.

Дальнейшее рассмотрение приводит поочередно то к системам, аналогичным системе (А), то — аналогичным системе (В). При этом для нахождения коэффициентов $b_{2(2l+1)}$ приходится решать уравнения степени $2l+1$, всегда имеющие действительный корень. После ряда таких рассмотрений, наконец, будут вычислены $b_3, b_4, b_3^{(2)}, b_1^{(2)}, b_0^{(2)}$. Оставшиеся вычисления выполняются совершенно аналогично вычислениям в примере с $n=4$.

Таким образом, если исходные коэффициенты были действительными, то пересчет приводит к также действительным коэффициентам.

В случае $n=2k+1$ применимость описанной схемы вытекает из соотношения $P_{2k+1} = x \tilde{P}_{2k} + a_0$, в котором

$$\tilde{P}_{2k} = x^{2k} + a_{2k}x^{2k-1} + \dots + a_2x + a_1.$$

Итак, приведенная схема позволяет вычислять полиномы n -ой степени с использованием $\lfloor (n+1)/2 \rfloor + \lfloor n/4 \rfloor$ умножений и $(n+1)$ сложений.

В заключение отметим работу Э. Белаги, которая будет скоро опубликована в сборнике „Математическое просвещение“. В ней проводится обобщение методов раздела 3 настоящей статьи на случай любого n и дается строгое доказательство невозможности построения схемы вычисления полиномов, использующей меньше, чем $\lfloor (n+1)/2 \rfloor$ умножений, или меньше, чем n сложений.

ЛИТЕРАТУРА

1. Studies in Mathematics and Mechanics, New York, Acad. Press Inc., 40, 1954.

Исследовательский физико-технический институт
при Горьковском университете

Поступила в редакцию.
5 мая 1958 г.

Схема сокращённого деления для цифровых вычислительных машин последовательного действия

Известия высших учебных заведений. Радиофизика, т.2, 1959, с. 322-323

СХЕМА СОКРАЩЕННОГО ДЕЛЕНИЯ ДЛЯ ЦИФРОВЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН ПОСЛЕДОВАТЕЛЬНОГО ДЕЙСТВИЯ

Ю. Л. Кетков

Идея, которая реализуется предлагаемой схемой, достаточно подробно описана в [1] и [2]. Однако для построения схемы сокращённого деления авторы этих работ требуют, чтобы делитель был нормализован (т. е. содержал единицу в старшем разряде) и чтобы деление выполнялось в варианте со сдвигом остатка влево.

Описываемая в данной статье схема предполагает представление чисел в прямых кодах с фиксированной в любом месте запятой и допускает деление как со сдвигом остатка, так и со сдвигом делителя, не требуя нормализации последнего.

В качестве арифметических регистров использованы рециркуляционные регистры, выполненные на магнитном барабане [3]. В цепь рециркуляции регистра включена цепочка динамических триггеров, с помощью которых можно изменить длину рециркуляционной петли и, тем самым, осуществлять сдвиг содержимого регистра сразу на несколько разрядов в ту или иную сторону.

В схеме имеются регистры делимого (P_1), делителя (P_2), результата (P_3) и управляющий регистр (P_4). Числа в регистрах циркулируют без знаков. Знак результата формируется специальной схемой и присваивается результату после выполнения операции. Одновременно с вводом чисел в регистры P_1 и P_2 в управляющий регистр P_4 вводится код, имеющий только одну единицу в разряде, соответствующем позиции 2^0 . Операция деления осуществляется в два этапа. На первом этапе производится «выравнивание порядков» — сдвиг делителя влево до тех пор, пока его старший разряд не окажется под старшим разрядом делимого. Одновременно происходит сдвиг влево на столько же разрядов управляющего кода в P_4 . Если делитель оказывается больше делимого, то «выравнивания порядков» не происходит. Следующий этап — собственно

деление, выполняемое по схеме без восстановления остатка. После каждого вычитания (сложения) производится сдвиг остатка (регистр P_1) влево на один разряд и одновременно сдвиг управляющего кода вправо на один разряд. Управляющий код используется для занесения единиц в соответствующие разряды частного (регистр P_2).

Описанная методика деления реализована на машине ГИФТИ [5, 4].

Значительное сокращение времени деления можно получить, если после получения очень малого остатка в результате очередного вычитания произвести сдвиг остатка и управляющего кода сразу на несколько разрядов.

Для реализации метода, основанного на этих соображениях, предлагается схема, изображенная на рис. 1.

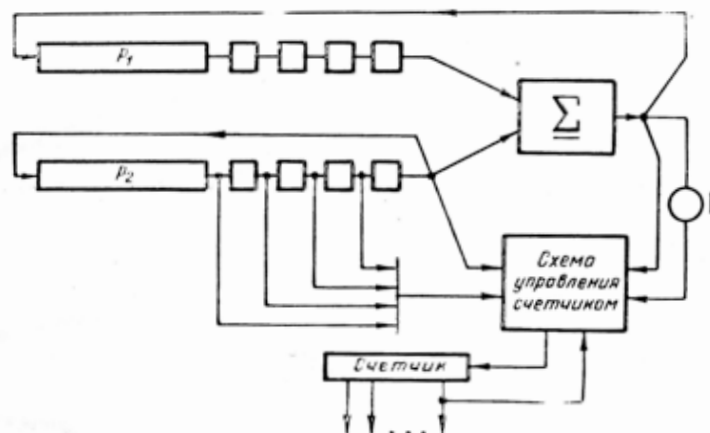


Рис. 1.

Счетчик определяет, на сколько разрядов следует произвести сдвиг после очередного вычитания. Емкость счетчика не должна превышать величины максимального возможного сдвига. Схема управления счетчиком реализует следующие правила.

1) Каждая единица инвертированного остатка прибавляет к счетчику единицу, если разрешено прибавление.

2) Каждая единица остатка сбрасывает счетчик и разрешает прибавление к нему.

3) Каждая единица делителя запрещает прибавление к счетчику при условии, что среди всех последующих разрядов делителя, просматриваемых на выходах предшествующих динамических триггеров, нет ни одной единицы.

4) После заполнения счетчика прибавление запрещается.

Разряды остатка, сбрасывающие счетчик и разрешающие прибавление к нему, следует подавать на схему управления счетчиком несколько задержанными.

В случае, если производится деление со сдвигом делителя, в приведенных правилах следует заменить роль разрядов остатка разрядами делителя, а разрядов делителя — разрядами остатка. Соответствующим образом меняется и схема.

ЛИТЕРАТУРА

1. Э. И. Клямко, Г. Д. Монахов, *Приборостроение*, 2, 9 (1957).
2. М. А. Карцев, *Арифметические устройства электронных цифровых машин*, изд. физ.-мат. лит., М., 151, 1958.
3. А. М. Гончаров, А. С. Гарантович, *Изв. высш. уч. зав. — Радиофизика*, 1, 2, 156 (1958).
4. М. Я. Эйнгорин, *Конференция Пути развития советского математического машиностроения и приборостроения, секция универсальных цифровых машин*, ч. 1, ВИНТИ, М., 1956.

Исследовательский физико-технический институт
при Горьковском университете

Поступила в редакцию
2 января 1959 г.

О существовании кодов Баркера

Известия высших учебных заведений. Радиофизика, т. III, с. 911-913

О СУЩЕСТВОВАНИИ КОДОВ БАРКЕРА

И. М. Ивалова, Ю. Л. Кетков, Т. С. Ямпольская

Построение кода Баркера длины n эквивалентно определению последовательности чисел a_j ($j=1, 2, \dots, n$), принимающих значения ± 1 и являющихся первой строкой симметричной матрицы A_{ij} :

	N_0	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8
a_1	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
$\frac{a_1 a_{2-1}}{a_2}$	$\frac{a_2 a_{3-1}}{a_3}$	$\frac{a_3 a_{4-1}}{a_4}$	$\frac{a_4 a_{5-1}}{a_5}$	$\frac{a_5 a_{6-1}}{a_6}$	$\frac{a_6 a_{7-1}}{a_7}$	$\frac{a_7 a_{8-1}}{a_8}$	$\frac{a_8 a_{9-1}}{a_9}$	a_{10}	a_{11}
$\frac{a_1 a_{3-1}}{a_3}$	$\frac{a_2 a_{4-1}}{a_4}$	$\frac{a_3 a_{5-1}}{a_5}$	$\frac{a_4 a_{6-1}}{a_6}$	$\frac{a_5 a_{7-1}}{a_7}$	$\frac{a_6 a_{8-1}}{a_8}$	$\frac{a_7 a_{9-1}}{a_9}$	a_{10}	a_{11}	a_{12}
$\frac{a_1 a_{4-1}}{a_4}$	$\frac{a_2 a_{5-1}}{a_5}$	$\frac{a_3 a_{6-1}}{a_6}$	$\frac{a_4 a_{7-1}}{a_7}$	$\frac{a_5 a_{8-1}}{a_8}$	$\frac{a_6 a_{9-1}}{a_9}$	a_{10}	a_{11}	a_{12}	a_{13}
$\frac{a_1 a_{5-1}}{a_5}$	$\frac{a_2 a_{6-1}}{a_6}$	$\frac{a_3 a_{7-1}}{a_7}$	$\frac{a_4 a_{8-1}}{a_8}$	$\frac{a_5 a_{9-1}}{a_9}$	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}
$\frac{a_1 a_{6-1}}{a_6}$	$\frac{a_2 a_{7-1}}{a_7}$	$\frac{a_3 a_{8-1}}{a_8}$	$\frac{a_4 a_{9-1}}{a_9}$	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
$\frac{a_1 a_{7-1}}{a_7}$	$\frac{a_2 a_{8-1}}{a_8}$	$\frac{a_3 a_{9-1}}{a_9}$	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}
$\frac{a_1 a_{8-1}}{a_8}$	$\frac{a_2 a_{9-1}}{a_9}$	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}
$\frac{a_1 a_{9-1}}{a_9}$	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}
a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}
a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}
a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}
a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	a_{22}
a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	a_{22}	a_{23}
a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	a_{22}	a_{23}	a_{24}
a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}
a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}
a_{18}	a_{19}	a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}
a_{19}	a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{28}
a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{28}	a_{29}
a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{28}	a_{29}	a_{30}
a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{28}	a_{29}	a_{30}	a_{31}
a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{28}	a_{29}	a_{30}	a_{31}	a_{32}
a_{24}	a_{25}	a_{26}	a_{27}	a_{28}	a_{29}	a_{30}	a_{31}	a_{32}	a_{33}
a_{25}	a_{26}	a_{27}	a_{28}	a_{29}	a_{30}	a_{31}	a_{32}	a_{33}	a_{34}
a_{26}	a_{27}	a_{28}	a_{29}	a_{30}	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}
a_{27}	a_{28}	a_{29}	a_{30}	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}
a_{28}	a_{29}	a_{30}	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}	a_{37}
a_{29}	a_{30}	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}	a_{37}	a_{38}
a_{30}	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}	a_{37}	a_{38}	a_{39}
a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}	a_{37}	a_{38}	a_{39}	a_{40}
a_{32}	a_{33}	a_{34}	a_{35}	a_{36}	a_{37}	a_{38}	a_{39}	a_{40}	a_{41}
a_{33}	a_{34}	a_{35}	a_{36}	a_{37}	a_{38}	a_{39}	a_{40}	a_{41}	a_{42}
a_{34}	a_{35}	a_{36}	a_{37}	a_{38}	a_{39}	a_{40}	a_{41}	a_{42}	a_{43}
a_{35}	a_{36}	a_{37}	a_{38}	a_{39}	a_{40}	a_{41}	a_{42}	a_{43}	a_{44}
a_{36}	a_{37}	a_{38}	a_{39}	a_{40}	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}
a_{37}	a_{38}	a_{39}	a_{40}	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	a_{46}
a_{38}	a_{39}	a_{40}	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	a_{46}	a_{47}
a_{39}	a_{40}	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	a_{46}	a_{47}	a_{48}
a_{40}	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	a_{46}	a_{47}	a_{48}	a_{49}
a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	a_{46}	a_{47}	a_{48}	a_{49}	a_{50}
a_{42}	a_{43}	a_{44}	a_{45}	a_{46}	a_{47}	a_{48}	a_{49}	a_{50}	a_{51}
a_{43}	a_{44}	a_{45}	a_{46}	a_{47}	a_{48}	a_{49}	a_{50}	a_{51}	a_{52}
a_{44}	a_{45}	a_{46}	a_{47}	a_{48}	a_{49}	a_{50}	a_{51}	a_{52}	a_{53}
a_{45}	a_{46}	a_{47}	a_{48}	a_{49}	a_{50}	a_{51}	a_{52}	a_{53}	a_{54}
a_{46}	a_{47}	a_{48}	a_{49}	a_{50}	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}
a_{47}	a_{48}	a_{49}	a_{50}	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}	a_{56}
a_{48}	a_{49}	a_{50}	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}	a_{56}	a_{57}
a_{49}	a_{50}	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}	a_{56}	a_{57}	a_{58}
a_{50}	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}	a_{56}	a_{57}	a_{58}	a_{59}
a_{51}	a_{52}	a_{53}	a_{54}	a_{55}	a_{56}	a_{57}	a_{58}	a_{59}	a_{60}
a_{52}	a_{53}	a_{54}	a_{55}	a_{56}	a_{57}	a_{58}	a_{59}	a_{60}	a_{61}
a_{53}	a_{54}	a_{55}	a_{56}	a_{57}	a_{58}	a_{59}	a_{60}	a_{61}	a_{62}
a_{54}	a_{55}	a_{56}	a_{57}	a_{58}	a_{59}	a_{60}	a_{61}	a_{62}	a_{63}
a_{55}	a_{56}	a_{57}	a_{58}	a_{59}	a_{60}	a_{61}	a_{62}	a_{63}	a_{64}
a_{56}	a_{57}	a_{58}	a_{59}	a_{60}	a_{61}	a_{62}	a_{63}	a_{64}	a_{65}
a_{57}	a_{58}	a_{59}	a_{60}	a_{61}	a_{62}	a_{63}	a_{64}	a_{65}	a_{66}
a_{58}	a_{59}	a_{60}	a_{61}	a_{62}	a_{63}	a_{64}	a_{65}	a_{66}	a_{67}
a_{59}	a_{60}	a_{61}	a_{62}	a_{63}	a_{64}	a_{65}	a_{66}	a_{67}	a_{68}
a_{60}	a_{61}	a_{62}	a_{63}	a_{64}	a_{65}	a_{66}	a_{67}	a_{68}	a_{69}
a_{61}	a_{62}	a_{63}	a_{64}	a_{65}	a_{66}	a_{67}	a_{68}	a_{69}	a_{70}
a_{62}	a_{63}	a_{64}	a_{65}	a_{66}	a_{67}	a_{68}	a_{69}	a_{70}	a_{71}
a_{63}	a_{64}	a_{65}	a_{66}	a_{67}	a_{68}	a_{69}	a_{70}	a_{71}	a_{72}
a_{64}	a_{65}	a_{66}	a_{67}	a_{68}	a_{69}	a_{70}	a_{71}	a_{72}	a_{73}
a_{65}	a_{66}	a_{67}	a_{68}	a_{69}	a_{70}	a_{71}	a_{72}	a_{73}	a_{74}
a_{66}	a_{67}	a_{68}	a_{69}	a_{70}	a_{71}	a_{72}	a_{73}	a_{74}	a_{75}
a_{67}	a_{68}	a_{69}	a_{70}	a_{71}	a_{72}	a_{73}	a_{74}	a_{75}	a_{76}
a_{68}	a_{69}	a_{70}	a_{71}	a_{72}	a_{73}	a_{74}	a_{75}	a_{76}	a_{77}
a_{69}	a_{70}	a_{71}	a_{72}	a_{73}	a_{74}	a_{75}	a_{76}	a_{77}	a_{78}
a_{70}	a_{71}	a_{72}	a_{73}	a_{74}	a_{75}	a_{76}	a_{77}	a_{78}	a_{79}
a_{71}	a_{72}	a_{73}	a_{74}	a_{75}	a_{76}	a_{77}	a_{78}	a_{79}	a_{80}
a_{72}	a_{73}	a_{74}	a_{75}	a_{76}	a_{77}	a_{78}	a_{79}	a_{80}	a_{81}
a_{73}	a_{74}	a_{75}	a_{76}	a_{77}	a_{78}	a_{79}	a_{80}	a_{81}	a_{82}
a_{74}	a_{75}	a_{76}	a_{77}	a_{78}	a_{79}	a_{80}	a_{81}	a_{82}	a_{83}
a_{75}	a_{76}	a_{77}	a_{78}	a_{79}	a_{80}	a_{81}	a_{82}	a_{83}	a_{84}
a_{76}	a_{77}	a_{78}	a_{79}	a_{80}	a_{81}	a_{82}	a_{83}	a_{84}	a_{85}
a_{77}	a_{78}	a_{79}	a_{80}	a_{81}	a_{82}	a_{83}	a_{84}	a_{85}	a_{86}
a_{78}	a_{79}	a_{80}	a_{81}	a_{82}	a_{83}	a_{84}	a_{85}	a_{86}	a_{87}
a_{79}	a_{80}	a_{81}	a_{82}	a_{83}	a_{84}	a_{85}	a_{86}	a_{87}	a_{88}
a_{80}	a_{81}	a_{82}	a_{83}	a_{84}	a_{85}	a_{86}	a_{87}	a_{88}	a_{89}
a_{81}	a_{82}	a_{83}	a_{84}	a_{85}	a_{86}	a_{87}	a_{88}	a_{89}	a_{90}
a_{82}	a_{83}	a_{84}	a_{85}	a_{86}	a_{87}	a_{88}	a_{89}	a_{90}	a_{91}
a_{83}	a_{84}	a_{85}	a_{86}	a_{87}	a_{88}	a_{89}	a_{90}	a_{91}	a_{92}
a_{84}	a_{85}	a_{86}	a_{87}	a_{88}	a_{8				

Матрицу A_{ij} , удовлетворяющую этим условиям, будем называть оптимальной.

В [1] показано, что коды Баркера, а следовательно, и оптимальные матрицы не существуют для $n = 4k \pm 1$ при $n > 13$. В настоящей заметке исследуется вопрос существования кодов Баркера для $n = 4k + 2$ и приводятся некоторые свойства оптимальных матриц для $n = 4k$.

Не теряя общности, можно ограничиться исследованием случая $a_n = \dots = 1$. Рассмотрим некоторые свойства оптимальной матрицы. Будем называть знаком диагонали произведение всех ее элементов. Выписывая выражения для знаков нулевых диагоналей, получаем:

$$\begin{aligned} \operatorname{sgn} A_1 &= -1 = a_2(a_n a_{n-1}); \\ \operatorname{sgn} A_2 &= +1 = \dots (a_2 a_{n-2})(a_1 a_{n-3}); \\ \operatorname{sgn} A_3 &= -1 = (a_3 a_{n-4})(a_6 a_{n-5}); \\ &\dots \end{aligned}$$

Таким образом,

$$(a_{2l+1} a_{n-2l})(a_{2l+2} a_{n-(2l+1)}) = -1. \tag{1}$$

Аналогичные выражения для знаков ненулевых диагоналей в силу (1) дают:

$$\operatorname{sgn} N_l = (-1)^l a_{2l+1} a_{n-2l}. \tag{2}$$

Свойство 1. В оптимальной матрице в случае $n = 4k$ имеет место соотношение:

$$\operatorname{sgn} N_l = \operatorname{sgn} N_{\frac{n-2}{2}-l}.$$

Действительно, в силу (2)

$$\operatorname{sgn} N_{\frac{n-2}{2}-l} = (-1)^{2k-l-1} a_{n-(2l+1)} a_{2l+2} = -(-1)^l a_{n-(2l+1)} a_{2l+2},$$

а из (1) следует, что

$$-a_{n-(2l+1)} a_{2l+2} = a_{2l+1} a_{n-2l}.$$

Поэтому знаки N_l и $N_{\frac{n-2}{2}-l}$ совпадают.

Свойство 2. Для оптимальной матрицы в случае $n = 4k$ справедливо равенство

$$S(N_l) + S\left(N_{\frac{n-2}{2}-l}\right) = 0.$$

Действительно, диагональ N_l содержит $2l+1$ элементов, из которых, в силу оптимальности матрицы, l элементов одного знака и $l+1$ противоположного знака; диагональ $N_{\frac{n-2}{2}-l}$ содержит соответственно $2k-l$ и $2k-l-1$ элементов противоположных знаков.

Возможны два случая: 1) либо диагональ N_l содержит l отрицательных членов и при этом $\operatorname{sgn} N_l = (-1)^l$ и $S(N_l) = 1$; в силу свойства 1

$$\operatorname{sgn} N_{\frac{n-2}{2}-l} = (-1)^l = (-1)^{2k-l}$$

и, следовательно, $S\left(N_{\frac{n-2}{2}-l}\right) = -1$; 2) либо диагональ N_l содержит $l+1$ отрицательных членов и при этом $\operatorname{sgn} N_l = (-1)^{l+1}$ и $S(N_l) = -1$; в силу свойства 1

$$\operatorname{sgn} N_{\frac{n-2}{2}-l} = (-1)^{l+1} = (-1)^{2k-l-1}$$

и, следовательно, $S\left(N_{\frac{n-2}{2}-l}\right) = 1$. Таким образом, в обоих случаях свойство 2 справедливо.

Теорема. Для $n = 4k + 2$ ($k = 1, 2, \dots$) оптимальной матрицы, а следовательно, и кода Баркера не существует.

Доказательство. Рассмотрим первую и последнюю нулевые диагонали A_1 и $A_{\frac{n-2}{2}}$. Очевидно, что

$$\operatorname{sgn} A_1 = -1 = a_2(a_1 a_{n-1});$$

тогда

$$\operatorname{sgn} A_{\frac{n-2}{2}} = a_{n-2}(a_{n-3} a_{n-1})(a_{n-4} a_{n-2}) \dots (a_{2l+2})(a_1 a_3) = a_{n-1} a_2 a_1 = -1.$$

Но, с другой стороны, диагональ $A_{\frac{n-2}{2}}$ содержит $4k+2-2=4k$ членов, и так как $A_{\frac{n-2}{2}}$ является нулевой диагональю, то в ней содержится $2k$ положительных и $2k$ отрицательных элементов. Поэтому

$$\text{sgn } A_{\frac{n-2}{2}} = (-1)^{2k} = 1.$$

Полученное противоречие доказывает теорему.

Путем проверки обнаружено, что коды Баркера не существуют для $n=8, 12, 16, 20$. В общем случае для $n=4k$ ($k=2, 3, \dots$) вопрос о существовании кодов Баркера остается открытым.

ЛИТЕРАТУРА

1. Ю. В. Поляк, Р. В. Мошетов, Научные труды радиотехн. ин-та АН СССР, т. 1, 124 (1959).

Научно-исследовательский физико-технический институт
при Горьковском университете

Поступила в редакцию
4 июня 1960 г.

Прошивка координатной матрицы запоминающего устройства на магнитных сердечниках

ИЗВЕСТИЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИИ

Том IV, № 1

РАДИОФИЗИКА

1961

ПРОШИВКА КООРДИНАТНОЙ МАТРИЦЫ ЗАПОМИНАЮЩЕГО УСТРОЙСТВА НА МАГНИТНЫХ СЕРДЕЧНИКАХ, РАБОТАЮЩЕЙ ПО ПРИНЦИПУ МНОГОКРАТНОГО СОВПАДЕНИЯ ТОКОВ

Ю. Л. Кетков

Обсуждается способ интерпретации прошивки координатной матрицы, состоящей из $n \times n$ сердечников и работающей по принципу P -кратного ($P \geq 2$) совпадения токов. Приводится способ построения такой прошивки для $P = c + 1$, где c — наименьший простой множитель числа n , отличный от 1.

Матрица, работающая по принципу P -кратного совпадения токов [1,2], прошивается P группами по n проводов в каждой группе, причем: 1) каждый провод прошивает n сердечников; 2) ни один сердечник не прошивается двумя проводами из одной и той же группы; 3) ни одна пара проводов разных групп не пересекается более, чем в одном сердечнике; 4) провода одной из групп прошивают матрицу по строкам, провода другой — по столбцам. В такой матрице ток в обмотках неоправиваемых сердечников не превосходит I/P (I — ток, необходимый для намагничивания одного сердечника). В случае $P = n-1$ влияние помехи в неоправиваемых сердечниках может быть сведено к нулю [2] пропускаясь во все невозбужденные провода противоположного тока $-I/n$ ($n+1$).

Если отбросить две тривиальные группы проводов, прошивающих матрицу по строкам и столбцам, то отыскание оставшихся групп можно свести к следующей математической задаче. Сопоставим каждому сердечнику матрицы элемент a_{ij} определителя n -го порядка (рис. 1).

Каждому проводу, удовлетворяющему выше приведенным условиям, сопоставим член определителя, составленный из тех a_{ij} , которые соответствуют сердечникам, прошитым данным проводом. Эти a_{ij} на самом деле составляют член определителя, так как отброшенные две тривиальные группы проводов запрещают оставшимся проводам проходить через какую-либо пару сердечников, расположенных в одной строке или в одном столбце.

Таким образом, отыскание оставшихся $P-2$ группы обрабатывающих проводов эквивалентно задаче выбора $n(P-2)$ членов определителя, выражения которых попарно содержат не более одного общего элемента a_{ij} (при этом следует рассматривать всевозможные пары членов определителя без учета их знаков). Назовем множество членов определителя, удовлетворяющих этому требованию, полным, если к нему нельзя присоединить ни одного члена определителя, не нарушив указанного требования. Число членов определителя, входящих в полное множество M , назовем размерностью полного множества $\rho(M)$. На примере $n=4$ можно убедиться, что существуют полные множества различных размерностей:



Рис. 1.

(смысл знака \oplus указан выше). Эта прошивка соответствует простому геометрическому построению. К исходному определителю надо приписать сбоку $n-1$ таких же определителей и провести через элементы первой строки исходного определителя $n-1$ группу параллельных прямых по n прямым в группе (рис. 2).

Теорема. Если n есть простое число, то выражения (1) составляют полное множество размерности $n(n-1)$.

Доказательство. Во-первых, убедимся в том, что выражения (1) при $j=1, 2, \dots, n$, $k=1, 2, \dots, n-1$ совпадают с выражениями для членов определителя (без учета их знаков). Для этого надо доказать, что в выражении (1) ни при каких j и k не может оказаться двух элементов из одного столбца. Другими словами, сравнение

$$\begin{aligned} j + mk &= j + lk \pmod{n} \\ (0 < l < m < n-1) \end{aligned} \quad (2)$$

не должно иметь решений при $j=1, 2, \dots, n$ и $k=1, 2, \dots, n-1$. Если бы это сравнение имело решение, то число $(m-l)k$ должно было бы делиться на n без остатка, что невозможно в силу простоты n . Таким образом, выражение (1) определяет $n(n-1)$ членов определителя.

Докажем теперь, что эти члены образуют полное множество. Для этого остается проверить выполнение условий 2) и 3). Другими словами, остается доказать, что сравнение

$$\begin{aligned} j_1 + mk &\equiv j_2 + mk \pmod{n} \\ (j_1 \neq j_2) \end{aligned} \quad (3)$$

и система сравнений

$$\begin{aligned} j_1 + mk_1 &= j_2 + mk_2 \pmod{n}; \\ j_1 + lk_1 &= j_2 + lk_2 \pmod{n} \\ (k_1 \neq k_2, \quad m \neq l) \end{aligned} \quad (4)$$

не имеют решений. Для сравнения (3) это очевидно, так как в противном случае число $j_1 - j_2$ должно было бы делиться на n без остатка, что невозможно. Система (4) также не имеет решения, так как в противном случае мы бы получили, что числа $j_1 - j_2 + m(k_1 - k_2)$ и $j_1 - j_2 + l(k_1 - k_2)$ должны делиться на n без остатка. Но тогда должно было бы делиться на n без остатка и число $(m-l)(k_1 - k_2)$, что невозможно в силу простоты n .

Итак, множество выражений (1) образует полное множество размерности $n(n-1)$.

Присоединяя к прошивке „веером“ отброшенные тривиальные группы, мы получаем $n+1$ группу опрашивающих проводов, удовлетворяющих условиям 1) ÷ 4).

Заметим, что прошивка „веером“ в случае составного n ($n = cd$; c — наименьший простой множитель числа n , отличный от 1) удовлетворяет условиям 1) ÷ 4) лишь для $k = 1, 2, \dots, c-1$.

В получающихся для $k \geq c$ группах опрашивающих проводов либо не выполняются условия 1) ÷ 4), либо каждый провод этих групп прошивает какую-либо пару сердечников, через которую уже проходит

провод одной из первых $c-1$ групп. Доказательство этого факта может быть легко установлено рассмотрением соответствующих сравнений.

ЛИТЕРАТУРА

1. R. C. Minnick and R. L. Ashenburt, *J. Appl. Phys.*, **26**, 575 (1955).
2. N. M. Blachman, *Trans. IRE*, **EC-5**, 1, 19 (1956).

Научно-исследовательский физико-технический институт
при Горьковском университете

Поступила в редакцию
27 мая 1960 г.

Об оптимальных методах нелинейной аппроксимации плоских кривых и системе автоматизации программирования для обработки геометрической информации. Автореферат

Министерство высшего и среднего специального образования РСФСР

Горьковский государственный университет им. Н. И. Лобачевского

НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ
ПРИКЛАДНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ
ПРИ ГГУ им. Н. И. ЛОБАЧЕВСКОГО

Ю. Л. КЕТКОВ

ОБ ОПТИМАЛЬНЫХ МЕТОДАХ НЕЛИНЕЙНОЙ
АППРОКСИМАЦИИ ПЛОСКИХ КРИВЫХ И СИСТЕМЕ
АВТОМАТИЗАЦИИ ПРОГРАММИРОВАНИЯ
ДЛЯ ОБРАБОТКИ ГЕОМЕТРИЧЕСКОЙ ИНФОРМАЦИИ

Автореферат
диссертации на соискание ученой степени
кандидата физико-математических наук

Научный руководитель — старший научный сотрудник,
кандидат физико-математических наук С. И. АЛЬБЕР.

ГОРЬКИЙ, 1966

Одной из актуальных проблем автоматизации программирования являются составление языка для решения задач технического проектирования и разработка соответствующих управляющих программ для автоматического перевода алгоритма на внутренний язык машины.

Настоящая работа посвящена исследованию и построению системы программирования, используемой для автоматизации проектных и конструкторских работ в судостроении, и исследованию возникающих при этом некоторых задач теории аппроксимации.

В первой главе диссертации рассматриваются задачи кусочно-полиномиальной аппроксимации. В отличие от классических подходов эти задачи являются по существу нелинейными, т. к. аппроксимирующие функции в этом случае образуют нелинейное многообразие. В § 1 приводится постановка общей задачи кусочно-полиномиальной аппроксимации и рассматриваются различные варианты задания исходной информации.

Пусть на отрезке $[a, b]$ задана кривая $y = f(x)$. Обозначим через G_n разбиение отрезка $[a, b]$ на n интервалов

$$a = x_0 < x_1 < x_2 < \dots < x_n = b$$

и внутри каждого интервала (x_i, x_{i+1}) соответствующий отрезок кривой будем аппроксимировать полиномом $P_{k,i}(x)$ степени не выше k . В результате аппроксимирующая функция $P_k(x, G_n)$ окажется составленной из кусков полиномов степени не выше k :

$$P_k(x, G_n) = \sum_{i=0}^{n-1} \varphi_{k,i}(x)$$

$$\varphi_{k,i}(x) = [1(x_{i+1}) - 1(x_i)] \cdot P_{k,i}(x)$$

$$1(x_i) = \begin{cases} 1 & \text{если } x > x_i \\ 0 & \text{если } x \leq x_i \end{cases}$$

Если разбиение G_n фиксировано, то при отсутствии ограничений на поведение $P_k(x, G_n)$ в узлах x_i нахождение ап-

проксимирующей кривой, наименее уклоняющейся от $f(x)$, сводится к хорошо известным классическим задачам аппроксимации внутри каждого из интервалов. Если же разбиение G отрезка $[a, b]$ заранее не фиксировать, то отыскание наилучшей аппроксимирующей кривой $P_k(x, G)$ приведет к существенно нелинейной задаче, т. к. искомые параметры (коэффициенты полиномов и узлы x_i) входят в аппроксимирующую функцию нелинейно и подпространство функций $\varphi_{k,i}(x)$ также нелинейно.

Наиболее полно исследуются (§§ 2 ÷ 9) вопросы кусочно-линейной аппроксимации плоских кривых. Аппроксимирующая ломаная считается оптимальной, если она приближает исходную кривую с заданной точностью и содержит при этом минимальное количество звеньев. В качестве уклонения ломаной от кривой рассматривается максимальное расстояние по нормали к звеньям ломаной.

В §§ 2, 3, 4 исследуется вопрос оптимальной аппроксимации кривой хордами (вписанной ломаной), который сводится к решению следующей задачи. Пусть (x, y) — точка на заданной кривой. Как далеко и в каком направлении можно выпустить из этой точки хорду, чтобы уклонение последней от кривой не превзошло заданной величины.

При достаточно высокой требуемой точности аппроксимации удалось доказать следующую теорему.

Теорема: пусть на заданном отрезке аппроксимируемой кривой, принадлежащей классу C^4 отсутствуют точки перегиба и монотонно меняется кривизна. Тогда для построения вписанной ломаной, уклоняющейся от кривой не более чем на ε , аппроксимация должна вестись в направлении уменьшения кривизны.

Если приближаемая кривая задана явным уравнением $y = f(x)$, то определение очередной вершины ломаной ведется по формулам

$$\begin{aligned} x_{n+1} &= x_n \pm h_1 \\ h_1 &= \sqrt{\frac{8\varepsilon \sqrt{1 + [f'(x_n)]^2}}{|f''(x_n)|}}. \end{aligned} \quad (1)$$

Если приближаемая кривая задана параметрическими уравнениями $x = x(t)$, $y = y(t)$, то соответствующие формулы имеют вид

$$\begin{aligned} x_{n+1} &= x(t_n \pm h_2) \\ y_{n+1} &= y(t_n \pm h_2) \end{aligned}$$

$$h_2 = \sqrt{\frac{8\varepsilon \sqrt{(x'_n)^2 + (y'_n)^2}}{|x'_n y''_n - x''_n y'_n|}} \quad (2)$$

В том случае, когда в качестве параметра выбирается длина дуги s , значение h_2 вычисляется по формуле

$$h_2 = \sqrt{\frac{8\varepsilon}{|K(s_n)|}},$$

где $K(s_n)$ — кривизна кривой в соответствующей вершине ломаной.

Если по указанным формулам аппроксимацию вести в направлении увеличения кривизны, то каждое звено получающейся ломаной будет уклоняться от кривой более, чем на ε . Таким образом, удается получить оценки сверху и снизу для числа звеньев оптимальной вписанной ломаной.

Полученные результаты были использованы (§ 5) для кусочно-линейной аппроксимации наиболее широко распространенных в технике профилей — кривых второго порядка. В частности, для них удалось показать, что в ряде случаев оценки приведенной теоремы остаются справедливыми и для произвольного уклонения ε . Численные эксперименты по приближению различных кривых второго порядка показали достаточно высокую эффективность предложенного алгоритма.

При достаточно высокой требуемой точности аппроксимации указанный алгоритм может быть использован для приближения плоских кривых касательными (описанными ломаными) или секущими (§§ 6, 7). Для этого достаточно построить ломаную, звенья которой параллельны найденным хордам и смещены в сторону кривой на расстояние d_i или $0,5 d_i$ (d_i — стрелка прогиба i -той хорды).

Для повышения точности формул (1) и (2) в работе используется метод неопределенных коэффициентов (§ 8). При этом величина h_i «шага» аппроксимации отыскивается в виде полинома

$$h_i = a_1 \sqrt{\varepsilon} + a_2 (\sqrt{\varepsilon})^2 + \dots + a_k (\sqrt{\varepsilon})^k.$$

В частности, для аппроксимации параболы соответствующие формулы для шагов h_i найдены при значениях $k = 2, 3$.

Далее (§§ 10, 11, 12) в работе рассматриваются вопросы оптимизации вычислений при кусочно-полиномиальной ап-

проксимации. В вычислительной практике для подсчета значения полинома широко используется схема Горнера

$$P_n(x) = ((a_n x + a_{n-1})x + a_{n-2})x + \dots)x + a_0,$$

использующая n сложений и $n + 1$ умножение. Задача улучшения схемы Горнера, впервые поставленная Дж. Толдом*, сводится к построению схемы вычисления полинома, использующей меньшее число операций. Оказалось, что этого можно достичь только после предварительной обработки исходных коэффициентов a_j . В связи с тем, что предварительная обработка для данного полинома выполняется однократно, то ее имеет смысл выполнять при необходимости в последующем многократного вычисления значений полинома. В работе показано, что невозможно построить схему вычисления полинома (даже с предварительной обработкой коэффициентов),

использующую меньше, чем $\left\lfloor \frac{n+1}{2} \right\rfloor$ умножений. Приводится

ряд схем для $n \leq 9$, позволяющих после предварительной обработки вычислять значение полинома, используя $n + 1$

сложение и $\left\lfloor \frac{n+1}{2} \right\rfloor$ умножений. Недостаток этих схем заклю-

чается в том, что иногда в результате предварительной обработки могут получиться комплексные числа, что сводит на нет в таких случаях все преимущества предлагаемых схем. Первая свободная от такого недостатка схема была опубликована

автором в 1958 г. Она позволяет вычислить значение полинома,

используя $n + 1$ сложение и $\left\lfloor \frac{n+1}{2} \right\rfloor + \left\lfloor \frac{n}{4} \right\rfloor$ умножений,

и для $n = 2k$ имеет следующий вид:

$$N_2 = x(b_0 + x)$$

$$N_4 = (N_2 + b_1 + x)(N_2 + b_2) + b_3$$

$$N_6 = N_2 N_4 + b_4 x + b_5$$

$$N_8 = (N_2 + b_6) N_6 + b_7$$

$$N_{10} = N_2 N_8 + b_8 x + b_9$$

$$\dots$$

$$N_{2k} = (N_2 + \bar{b}_k b_{2k-2}) N_{2k-2} + \bar{b}_k b_{2k-2} x + b_{2k-1}$$

* Todd J. Commons, on Pure and Appl. Math., 8, № 1, 1955, (русский перевод: Дж. Толд. Мотивы для работы в области численного анализа. «Матем. просвещение», вып. 1, 1957).

здесь

$$\bar{\delta}_k = 1 - \delta_k = \begin{cases} 1 & \text{если } k\text{—четное} \\ 0 & \text{если } k\text{—нечетное} \end{cases}.$$

Предварительная обработка сводится к определению коэффициентов b_j из условия $P_{2k} = N_{2k}$. Доказано, что если исходные коэффициенты a_i были действительными, то пересчитанные коэффициенты будут также действительными.

Позднее В. Я. Паном* были предложены более экономичные схемы, использующие для вычисления полинома $n + 1$ сложение и $\left[\frac{n+1}{2} \right]$ умножений.

Вторая глава посвящена вопросам разработки систем автоматического программирования, использованных, в частности, для автоматизации проектных и конструкторских работ в судо- и авиастроении. В качестве входного языка разработанные системы используют матричные программы (§ 1), записываемые в виде некоторой матрицы Π , состоящей из трех частей:

$$\Pi = \left\| \begin{array}{c} (A) \\ (R) \\ (B) \end{array} \right\|$$

Здесь A — прямоугольная матрица размерности $m \times n$, R — вектор-столбец размерности m , B — прямоугольная матрица размерности $m \times l$ (величины n, l определяются особенностями конкретного класса задач, m — объемом решаемой задачи).

Элементы матрицы A рассматриваются как набор аргументов (исходные данные и промежуточные результаты вычислений) или сведений об их расположении в таблицах, информационных списках или в памяти машины; элементы вектор-столбца R — как коды псевдоопераций, выполняемые над элементами соответствующих строк матрицы A ; элементы матрицы B — как сведения о результатах вычислений и дальнейшем их использовании. В частности, элементами матрицы B могут служить тройки чисел (i, j, k) , определяющие индекс результата вычислений (в результате операции над элемента-

* В. Я. Пан. О способах вычисления значения многочленов. «Успехи матем. наук», 1966, т. XXI, вып. 1 (127).

ми одной строки может быть получено несколько величин, которые для различения должны быть снабжены метками-индексами) и индексы элементов матрицы A (например, номера—строки и столбца), которые используют данный результат при последующих вычислениях.

В случае удачного выбора структуры матричной программы и соответствующего подбора псевдоопераций для некоторых классов задач удается создать весьма компактную форму записи алгоритма решения задачи. Перевод матричных программ в код любой универсальной ЦВМ может быть выполнен с помощью управляющих программ интерпретирующего или компилирующего типов в зависимости от объема исходной информации, объема запоминающих устройств и быстродействия машины.

В качестве одной из возможных реализаций метода матричного программирования в § 2 приводится описание интерпретирующих систем УИС-РГМ и ИС-РГВ, предназначенных для автоматизации плазовых работ в судо- и авиастроении.

Каждая из разработанных интерпретирующих систем состоит из библиотеки стандартных подпрограмм (БСП) и программы интерпретации. БСП предназначена для реализации на ЭВМ набора кодов псевдоопераций, соответствующих геометрическим построениям, используемым при проектировании поверхности. Набор кодов псевдоопераций включает всевозможные построения, опирающиеся на аппроксимацию кривых дугами окружностей и отрезками прямых. Однако, без всякого труда его можно пополнить и другими операциями, позволяющими использовать в построениях любые более сложные кривые. По своей структуре каждая из библиотек стандартных подпрограмм (в интерпретирующих системах УИС-РГМ и ИС-РГВ используется соответственно БСП-1 и БСП-2, приведенные в приложениях 1 и 2) может быть разбита на четыре группы задач: определение координат точки по данной информации, определение недостающих параметров в уравнениях прямой и окружности, выполнение арифметических и логических операций, специальные операции для реализации специфически плазовых построений (расчет ватерлиний, батоксов, рыбин, развертка листов и т. п.).

Далее в §§ 4 ÷ 7 описываются состав исходной информации по расчету сечений поверхности, развертке листов, приводятся примеры по составлению матричных программ и их перекодировке для ввода в машину. Организация и работа интерпретирующих систем описаны в § 8. Здесь же приводятся описания наиболее интересных специальных подпрограмм,

связанных с организацией циклов и условных переходов в матричных программах, с пересечением пространственных кривых на поверхности, с разверткой листов судовой поверхности на плоскость.

Разработанные интерпретирующие системы находятся в эксплуатации с 1963 г. и в настоящее время внедрены в ряде проектных и конструкторских организаций судостроительной промышленности. С их помощью были проведены расчеты по составлению плазовых книг и развертке листов обшивки для ряда судов различных типов: земснаряд, нефте-рудовоз, цементовоз, самоходная баржа, морские сухогрузные суда и ряд других. На основании трехлетнего опыта эксплуатации указанных систем можно сделать следующие выводы.

Матричное программирование является одним из методов автоматизации программирования с использованием макроинструкций. Оно позволяет существенным образом уменьшить объем входной информации при автоматизации проектных и конструкторских работ, связанных с обработкой геометрической информации. Изменение набора стандартных подпрограмм делает матричное программирование очень гибким и позволяет легко приспособить разработанные интерпретирующие системы для других методов аналитического проектирования сложных криволинейных поверхностей (например, с использованием кривых n -го порядка).

Применение систем автоматизации программирования обработки геометрической информации позволяет эффективно использовать вычислительную технику для внедрения аналитических методов проектирования сложных криволинейных поверхностей и сократить сроки подготовки плазовой документации и постройки судов за счет широкого использования станков с цифровым программным управлением.

Дальнейшее совершенствование аналитических способов задания сложных криволинейных поверхностей выдвигает ряд новых математических проблем. Одной из них, в частности, является задача кусочно-полиномиальной аппроксимации плоских кривых и поверхностей.

Основные результаты работы опубликованы в следующих статьях:

1. Ю. Л. Кетков. Об одном способе вычисления полиномов на математических машинах. Известия ВУЗов, Радиофизика, т. 1, № 4, 1958.
2. Ю. Л. Кетков. Об оптимальных методах кусочно-линейной аппроксимации. Известия ВУЗов, Радиофизика, т. 9, № 5, 1966.
3. Ю. Л. Кетков. Автоматизация проектирования поверхностей корпусов судов. Сб. статей «Автоматизация технологического проектирования при помощи ЭВМ», Машиностроение, М., 1966.

Результаты работы докладывались на следующих конференциях и совещаниях:

1. Всесоюзная конференция по вычислительной технике, М., 1959.
2. Техническое совещание по обмену опытом в области проведения плазово-шаблонных работ, НИИТ, М., 1964.
3. Техническое совещание Государственного Комитета по судостроению СССР, М., 1964.
4. Семинар МАИ «Кибернетика графики», М., 1964.
5. Семинары отдела математической физики и методов вычислений НИИ ПМК, Горький, 1965.

Об оптимальных методах кусочно-линейной аппроксимации

ИЗВЕСТИЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ

Том IX, № 6

РАДИОФИЗИКА

1966

УДК 518.43

ОБ ОПТИМАЛЬНЫХ МЕТОДАХ КУСОЧНО-ЛИНЕЙНОЙ АППРОКСИМАЦИИ

Ю. Л. Кетков

Рассматриваются методы оптимальной кусочно-линейной аппроксимации произвольных плоских кривых. Аппроксимирующая ломаная считается оптимальной, если она приближает кривую с наперед заданной точностью и содержит минимальное количество звеньев. Предлагаемая методика позволяет строить ломаные, достаточно близкие к оптимальным, и получать оценки сверху и снизу для количества звеньев оптимальной ломаной при достаточно высокой точности аппроксимации.

Решение многих задач связано с аппроксимацией различных кривых — нелинейных характеристик систем автоматического регулирования [1], траекторий движения режущего инструмента при обработке деталей на станках с цифровым программным управлением [2], контуров сложных криволинейных поверхностей при проектировании в судостроении [3] и т. п. В настоящее время хорошо разработаны методы теории и практики приближения [4] в тех случаях, когда аппроксимирующие функции образуют линейное подпространство в общем пространстве приближаемых функций. Однако эти методы нельзя использовать тогда, когда подпространство аппроксимирующих функций нелинейно. В частности, к такой нелинейной задаче приводит замена кривых малыми прямолинейными или криволинейными отрезками, концы которых заранее не фиксируются.

При решении такой задачи возможны два подхода. В первом случае при заданной точности аппроксимации требуется построить приближающую функцию, состоящую из минимального числа звеньев. Во втором случае при заданном числе звеньев требуется достичь наилучшей точности. В настоящей статье рассматриваются методы оптимальной кусочно-линейной аппроксимации, приводятся оценки сверху и снизу минимального числа звеньев и алгоритмы построения ломаных, достаточно близких к оптимальным.

1. ПОСТАНОВКА ЗАДАЧИ

Наибольшее распространение при кусочно-линейной аппроксимации кривых имеет замена их хордами, касательными или секущими, проведенными через соответствующие точки заменяемой кривой. Мы будем исследовать, в основном, приближение кривых хордами и при достаточно высокой точности аппроксимации используем полученные результаты для приближения кривых касательными или секущими.

Пусть F — отрезок кривой, не содержащей точек перегиба и для определенности обращенной вогнутостью вниз (рис. 1). Заданный отрезок кривой аппроксимируется вписанной ломаной линией L_n (n — число звеньев ломаной) таким образом, чтобы стрелка прогиба для каждого звена не превышала наперед заданную точность. Очевидно,

что при достаточно большом числе звеньев ломаной заданная точность может быть достигнута. Но при этом возникает два вопроса: а) как оценить минимальное число n_0 звеньев ломаной, аппроксимирующей отрезок кривой F с заданной точностью, и б) как построить ломаную линию с числом звеньев, достаточно близким к n_0 .

Для решения поставленных задач рассмотрим почти эквивалентную задачу. Пусть (x_0, y_0) — точка на кривой F (рис. 2). Требуется выбрать вторую точку (x_1, y_1) на кривой, такую, чтобы хорда, проходящая через эти две точки, отклонялась от кривой F не более чем на ϵ .

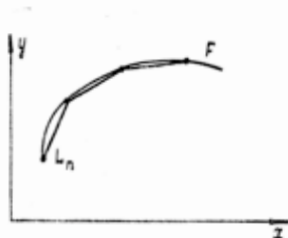


Рис. 1. Аппроксимация кривой хордами.

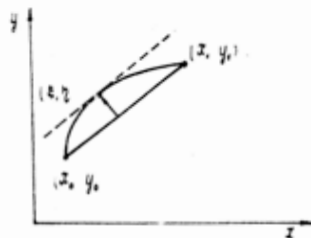


Рис. 2.

Обозначим через (ξ, η) точку кривой F , максимально удаленную от хорды, соединяющей точки (x_0, y_0) и (x_1, y_1) . Так как кривая F не содержит точек перегиба, то касательная к кривой в точке (ξ, η) должна быть параллельна хорде.

Пусть уравнение хорды и указанной касательной соответственно

$$y = kx + b; \quad (1)$$

$$Y = kX + B. \quad (2)$$

Здесь

$$k = \frac{y_1 - y_0}{x_1 - x_0}, \quad b = y_0 - kx_0, \quad B = \eta - k\xi.$$

Такая форма задания хорды и касательной не ограничивает общности, так как в случае вертикальных прямых ($k=b=\infty$) этого ограничения можно избежать заменой координатных осей.

Расстояние между хордой (1) и касательной (2) выражается формулой

$$d = \frac{B - b}{\sqrt{1 + k^2}}. \quad (3)$$

Так как эта величина и является максимальным отклонением кривой F от хорды, то для решения сформулированной задачи надо решить уравнение

$$d = \frac{B - b}{\sqrt{1 + k^2}} = \epsilon. \quad (4)$$

Сейчас мы приступим к приближенному решению уравнения (4), с помощью которого и будут получены ответы на поставленные вопросы. Затем эти результаты будут использованы для кусочно-линейной аппроксимации кривых второго порядка.

2. АППРОКСИМАЦИЯ КРИВОЙ, ЗАДАННОЙ В ЯВНОМ ВИДЕ $y=f(x)$

Пусть $x_1 = x_0 + h$, где h — пока неизвестная величина, которую надо определить. Тогда

$$k(h) = \frac{f(x_0+h) - f(x_0)}{h} = \frac{\Delta f(x_0)}{h},$$

$$b(h) = f(x_0) - x_0 k(h),$$

$$f'(\xi) = k(h),$$

$$B(h) = f(\xi) - \xi k(h),$$

$$d(x_0+h) = \frac{B(h) - b(h)}{(1+k^2(h))^{1/2}}.$$

Для приближенного решения уравнения (4) разложим функцию $d(h)$ в ряд Тейлора в окрестности $h=0$. Опуская довольно громоздкие промежуточные выкладки, приведем окончательный результат:

$$d(x_0+h) \approx \frac{1}{2!} d''(0) h^2 + \frac{1}{3!} d'''(0) h^3 + \dots, \quad (5)$$

где

$$d''(0) = -f''(x_0)/4 [1 + [f'(x_0)]^2]^{1/2},$$

$$d'''(0) = -\frac{3[f'''(x_0) (1 + [f'(x_0)]^2) - f'(x_0)[f''(x_0)]^2]}{8(1 + [f'(x_0)]^2)^{3/2}}.$$

Если ε достаточно мало, то соответственно мало и h . Поэтому в качестве приближенного решения уравнения (4) можно взять

$$h_1 = (2\varepsilon / |d''(0)|)^{1/2} = [8\varepsilon(1 + [f'(x_0)]^2)^{1/2} / |f''(x_0)|]^{1/2}. \quad (6)$$

Таким образом, при достаточно малом ε стрелка прогиба выражается следующей приближенной формулой:

$$d(x_0 \pm h_1) \approx \varepsilon \pm \frac{1}{3!} d'''(0) h_1^3.$$

Предположим далее, что на заданном отрезке кривой выражение $d'''(0)$ не меняет знака (геометрический смысл этого предположения будет объяснен ниже) и для определенности положительно. Тогда при аппроксимации «вперед»

$$x_{n+1} = x_n + h_1,$$

$$h_1 = [8\varepsilon(1 + [f'(x_n)]^2)^{1/2} / |f''(x_n)|]^{1/2}$$

уклонение ломаной от кривой всегда будет превосходить ε , а при аппроксимации «назад»

$$x_{n+1}^- = x_n^- - h_1,$$

$$h_1 = [8\varepsilon(1 + [f'(x_n)]^2)^{1/2} / |f''(x_n)|]^{1/2}$$

уклонение никогда не будет превышать ε .

3. АППРОКСИМАЦИЯ КРИВОЙ, ЗАДАННОЙ В ПАРАМЕТРИЧЕСКОЙ ФОРМЕ

$$x = x(t), \quad y = y(t)$$

Будем решать аналогичную задачу. Пусть $x_0 = x(t_0)$, $y_0 = y(t_0)$, $x_1 = x(t_0 + h)$, $y_1 = y(t_0 + h)$. Как далеко можно шагнуть по t , чтобы максимальное отклонение кривой от хорды не превысило ε ?

Максимальное отклонение кривой от хорды достигается в точке $(x(t_0 + \xi), y(t_0 + \xi))$, для которой

$$y'(t_0 + \xi)/x'(t_0 + \xi) = k(h) = [y(t_0 + h) - y(t_0)]/[x(t_0 + h) - x(t_0)].$$

Это отклонение выражается формулой

$$d(t_0 + h) = [B(h) - b(h)]/(1 + k^2(h))^{1/2}.$$

Здесь

$$b(h) = y(t_0) - x(t_0) k(h),$$

$$B(h) = y(t_0 + \xi) - x(t_0 + \xi) k(h).$$

Разложение функции $d(h)$ в ряд Тейлора в окрестности $h = 0$ имеет вид

$$d(h) = \frac{1}{2!} d''(0) h^2 + \frac{1}{3!} d'''(0) h^3 + \dots, \quad (7)$$

где

$$d''(0) = -(x'_0 y_0 - x_0 y'_0) / 4 [(x'_0)^2 + (y'_0)^2]^{1/2},$$

$$d'''(0) = -3 [(x'_0 y_0 - x_0 y'_0) [(x'_0)^2 + (y'_0)^2] - (x'_0 y'_0 - x_0 y_0) \times \\ \times (x'_0 x_0 + y'_0 y_0)] 8^{-1} [(x'_0)^2 + (y'_0)^2]^{-3/2}.$$

Индекс «0» внизу здесь означает, что все производные берутся в точке $t = t_0$.

Итак, при достаточно малом ε в качестве приближенного решения уравнения (4) можно взять

$$h_2 = (2\varepsilon / |d''(0)|)^{1/2} = [8\varepsilon [(x'_0)^2 + (y'_0)^2]^{1/2} |x'_0 y_0 - x_0 y'_0|^{-1}]^{1/2}. \quad (8)$$

Наиболее простой смысл разложение (7) приобретает в том случае, когда в качестве параметра t выбирается длина дуги s кривой. В этом случае

$$(x')^2 + (y')^2 = 1,$$

$$d''(0) = \frac{1}{4} K(s),$$

$$d'''(0) = \frac{3}{8} K'(s),$$

($K(s)$ — кривизна) и разложение (7) приобретает инвариантный характер:

$$d(s+h) = \frac{1}{8} K(s) h^2 + \frac{1}{16} K'(s) h^3 + \dots \quad (9)$$

Приближенное решение (8) тогда переписывается в форме

$$h_2 = (8\varepsilon / |K(s)|)^{1/2} = (8\varepsilon / |R(s)|)^{1/2}, \quad (10)$$

где $R(s)$ — радиус кривизны в данной точке кривой.

Заметим, что смысл постоянства знака $d'''(0)$ эквивалентен монотонности изменения кривизны $K(s)$ вдоль кривой.

4. КУСОЧНО-ЛИНЕЙНАЯ АППРОКСИМАЦИЯ КРИВЫХ ВТОРОГО ПОРЯДКА

Применим полученные результаты к кусочно-линейной аппроксимации кривых второго порядка.

Пример 1. Если рассматривать каноническое уравнение параболы $y^2 = 2px$ и из соображений симметрии аппроксимировать только ее верхнюю ветвь ($y > 0$), то максимальное уклонение хорды, проходящей через две точки (x_0, y_0) и (x_1, y_1) параболы, выражается формулой

$$d = (y_1 - y_0)^2 / 4[4p^2 + (y_1 + y_0)^2]^{1/2}$$

и достигается в точке (ξ, η) параболы

$$\xi = (y_0 + y_1)^2 / 8p, \quad \eta = (y_0 + y_1) / 2.$$

Отсюда видно, что смещение по параболе удобнее вести вдоль координаты y ($y_1 = y_0 \pm h$). При этом приближенное решение уравнения (4)

$$d(y_0 \pm h) = h^2 / 4[4p^2 + (2y_0 \pm h)^2]^{1/2} = \varepsilon \tag{11}$$

может быть получено из общей формулы (6) в виде

$$h_1 = [8\varepsilon(p^2 + y_0^2)^{1/2}]^{1/2}.$$

Нетрудно убедиться в том, что при любом ε аппроксимация «вперед» с шагом h_1 удовлетворяет требованиям точности, т. е.

$$d(y + h_1) < \varepsilon.$$

При аппроксимации «назад» с шагом h_1 требования точности соблюдаются лишь при выполнении ограничения

$$\varepsilon < 2y/(p^2 + y^2)^{1/2},$$

т. е. в том случае, если приближаемый участок параболы не лежит слишком близко к ее вершине.

В связи с тем, что уравнение (4) для параболы имеет сравнительно простой вид, можно получить более точное решение методом неопределенных коэффициентов. Для этого решение уравнения (11) ищется в виде

$$\tilde{h}_k = a_1 \delta + a_2 \delta^2 + \dots + a_k \delta^k \quad (\delta = \sqrt{\varepsilon}).$$

Приведем без вывода несколько найденных таким способом решений:

$$\tilde{h}_1 = [8(p^2 + y^2)^{1/2}]^{1/2} \delta,$$

$$\tilde{h}_2 = \tilde{h}_1 + 2y(p^2 + y^2)^{-1/2} \delta^2,$$

$$\tilde{h}_3 = \tilde{h}_2 + (2p^2 + y^2)[2(p^2 + y^2)^{5/2}]^{-1/2} \delta^3.$$

Пример 2. Если рассматривать уравнение эллипса в параметрической форме $x = a \cos t$, $y = b \sin t$ и из соображений симметрии ограничиться только одной его четвертью ($0 \leq t \leq \pi/2$), то максимальное уклонение хорды от эллипса на отрезке $(t, t \pm h)$ выражается формулой

$$d(t \pm h) = ab(1 - \cos h/2) [a^2 \sin^2(t \pm h/2) + b^2 \cos^2(t \pm h/2)]^{-1/2}$$

и достигается в точке $t \pm h/2$. Приближенное решение уравнения (4) при этом получается в виде

$$h_2 = [8\varepsilon(a^2 \sin^2 t + b^2 \cos^2 t)^{1/2}/ab]^{1/2}.$$

Нетрудно убедиться в том, что при любом ε аппроксимация «вперед» с шагом h_2 удовлетворяет требованиям точности. Это позволяет получить оценку числа звеньев оптимальной ломаной для эллипса сверху.

Оценка уклонения ломаной, построенной при аппроксимации эллипса «назад» с шагом h_2 , в общем случае (ε не мало) довольно затруднительна. Однако сразу можно заметить, что далеко не для каждого эллипса справедливо неравенство

$$d(t-h_2) > \varepsilon.$$

Действительно, в частном случае для окружности ($a = b = R$)

$$d(t+h_2) = d(t-h_2) = R(1-\cos h_2/2) < \varepsilon;$$

это же справедливо по непрерывности для семейства эллипсов, достаточно близких к окружности.

Пример 3. По аналогии с эллипсом можно провести кусочно-линейную аппроксимацию гиперболы, заданной в параметрической форме $x = a \operatorname{ch} t$, $y = b \operatorname{sh} t$ ($0 < t < \infty$). Для приближенного решения уравнения (4)

$$d(t \pm h) = ab[\operatorname{ch}(h/2) - 1] [a^2 \operatorname{sh}^2(t \pm h/2) + b^2 \operatorname{ch}^2(t \pm h/2)]^{-1/2} = \varepsilon$$

здесь получено выражение

$$h_3 = [8\varepsilon(a^2 \operatorname{sh}^2 t + b^2 \operatorname{ch}^2 t)^{1/2}/ab]^{1/2}.$$

Аппроксимация гиперболы «назад» с шагом h_3 при любом ε приводит к ломаной, уклоняющейся от гиперболы всегда на расстояние, большее ε ; таким образом получается оценка снизу для количества звеньев оптимальной ломаной. Оценка поведения ломаной при аппроксимации гиперболы «вперед» с шагом h_3 в общем случае (ε не мало) довольно затруднительна.

5. АЛГОРИТМ АППРОКСИМАЦИИ ПРОИЗВОЛЬНОЙ КРИВОЙ ХОРДАМИ

Если заданная точность аппроксимации ε достаточно высока, то для построения вписанной ломаной, близкой к оптимальной (в смысле количества звеньев), может быть предложен следующий алгоритм.

Заданную кривую следует разбить на отрезки, не содержащие точек перегиба. Для построения ломаной, уклоняющейся от кривой не более чем на ε , аппроксимация должна вестись на каждом из этих отрезков в направлении уменьшения кривизны, так как именно в этом случае знаки первых двух членов в разложениях (5), (7), (9) противоположны. Таким образом, на каждом из выделенных отрезков «направление» аппроксимации должно меняться на противоположное при каждом переходе через точку экстремума кривизны. Например, для эллип-

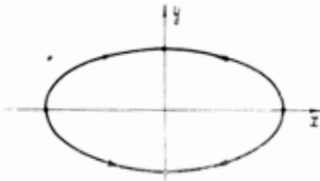


Рис. 3. «Направления» аппроксимации для эллипса.

са аппроксимация должна вестись от вершин, лежащих на большой полуоси, к вершинам, лежащим на малой полуоси (см. рис. 3).

«Шаг» аппроксимации, в зависимости от формы задания кривой, выбирается по одной из формул (6), (8) или (10). Получающееся при этом число звеньев аппроксимирующей ломаной дает оценку сверху для количества звеньев n_0 оптимальной ломаной.

Если аппроксимацию на выделенных отрезках кривой вести в направлении увеличения кривизны с одним из найденных шагов (6), (8) или (10), то каждое звено получающейся ломаной всегда уклоняется от кривой более чем на ε . Это позволяет получить оценку для оптимального значения n_0 снизу.

6. АЛГОРИТМ АППРОКСИМАЦИИ ПРОИЗВОЛЬНОЙ КРИВОЙ КАСАТЕЛЬНЫМИ ИЛИ СЕКУЩИМИ

При достаточно высокой заданной точности аппроксимации (ε мало) предложенный выше алгоритм может быть использован для аппроксимации кривых касательными или секущими. Для этого достаточно построить ломаную, звенья которой параллельны найденным хордам и смещены в сторону кривой на расстояние d_i или $0,5 d_i$ (d_i — стрелка прогиба для i -ой хорды). Если $d_i = \varepsilon$, то построенные отрезки касательных почти везде будут уклоняться от кривой не более чем на ε , за исключением стыков ломаной (см. заштрихованную область на рис. 4), в которых ширина «ленты аппроксимации» достигает максимальной величины:

$$d_i = \varepsilon / \cos(\varphi_i/2).$$

Но так как при достаточно малых ε величина φ_i , равная разности углов наклона ломаной, близка к нулю, а изложенный алгоритм построения хорд всегда приводит к $d_i < \varepsilon$, то указанным исключением можно пренебречь.

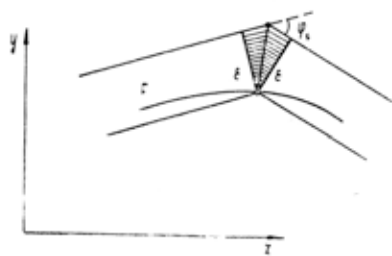


Рис. 4. Аппроксимация кривой касательными.

Предлагаемая методика кусочно-линейной аппроксимации произвольных кривых позволяет строить ломаные, достаточно близкие к оптимальным. Одновременно удается получить оценки сверху и снизу для количества звеньев n_0 оптимальной ломаной при достаточно высокой точности аппроксимации (ε мало). Применение указанных алгоритмов для аппроксимации наиболее широко используемых профилей (кривых второго порядка) позволяет получить некоторые оценки и в тех случаях, когда заданная точность не является достаточно малой. Предлагаемые алгоритмы описываются довольно простыми рекуррентными соотношениями и поэтому легко программируются при вычислениях на ЭВМ.

ЛИТЕРАТУРА

1. В. В. Солодовников, Ю. И. Топчиев, Г. В. Крутиков, Частотный метод построения переходных процессов, Гостехиздат, М., 1955.
2. В. А. Барун, А. А. Будинский, Станки с программным управлением и программирование обработки, изд. Машиностроение, М.—Л., 1965.

3. М. Н. Рейнов, В. И. Брегман, В. М. Москаленко, Э. М. Нахимович, Е. Ю. Петров, Н. Л. Мошенский, Е. М. Аксенов. Судостроительные расчеты на электронных вычислительных машинах, изд. Судостроение, Л., 1964.
4. Н. И. Ахиезер. Лекции по теории аппроксимации, изд. Наука, М., 1965.

Научно-исследовательский институт прикладной математики и кибернетики при Горьковском университете Поступила в редакцию
15 февраля 1966 г.

ON OPTIMUM METHODS OF PIECE-LINEAR APPROXIMATION

Yu. L. Ketkov

Methods of an optimum piece-linear approximation of the arbitrary plane curves are considered. An approximating curve is to be optimum if it gives the curve with predetermined accuracy and contains the minimum number of sections. The method presented in the article permits to plot the broken lines sufficiently close to the optimum ones and to obtain the estimations „above“ and „below“ for the number of sections of the optimum broken line at the sufficiently high accuracy of approximation.

Автоматизация проектирования поверхностей корпусов судов при помощи ЭВМ

АВТОМАТИЗАЦИЯ ТЕХНОЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ ПРИ ПОМОЩИ ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

СБОРНИК СТАТЕЙ

под редакцией
канд. техн. наук А. М. ГИЛЬМАНА



ИЗДАТЕЛЬСТВО «МАШИНОСТРОЕНИЕ»
Москва — 1966

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ ПОВЕРХНОСТЕЙ КОРПУСОВ СУДОВ ПРИ ПОМОЩИ ЭВМ

Развитие аналитических методов задания обводов судовой поверхности открывает широкие возможности использования современной вычислительной техники для автоматизации ряда проектных и конструкторских работ. Существующая технология проектирования судовой поверхности требует значительной доработки теоретического чертежа в заводских условиях на плазе.

Графическая разбивка и согласование теоретических линий обвода корпуса в трех проекциях («корпус», «полуширота», «бок») является одной из трудоемких операций, которая предъявляет очень высокие требования к квалификации исполнителей и качеству используемых материалов и инструментов. Значительный субъективизм в оценке плавности линий приводит к существенным отклонениям между плазовыми книгами, составленными на разных судостроительных заводах по одному и тому же проекту.

Аналитическое задание судовой поверхности позволяет устранить этот недостаток, делает возможным разработку единой документации, не требующей дальнейшей доработки. Использование электронной вычислительной техники при этом существенно облегчает выполнение масштабной разбивки, используемой при фотопроекционных методах разметки материала, изготовлении гибочных и проверочных шаблонов, при газовой вырезке деталей и т. д.

Возможность аналитического определения размеров деталей и различных конструкций корпуса, зависящих от формы судовой поверхности, создает предпосылки для широкого внедрения программно управляемых станков.

В настоящее время используется несколько различных методов аналитического задания судовых поверхностей. Общим у всех является то, что на поверхности некоторым образом задается базовая сетка линий — система криволинейных координат [4]. Одни линии, как правило, имеют продольное направление, другие — поперечное. Разница в методах в основном характеризуется формой задания этих линий. Например, линии базовой сетки могут быть представлены аналитическими функциями (преимущественно сопряженными отрезками квадратичных и кубических парабол), аппроксимирующими полиномами, совокупностью сопрягающихся прямых и окружностей (радиусографический метод), совокупностью точек на лекальных кривых.

С помощью введенной системы координат по известным формулам аналитической геометрии могут быть определены координаты

наты любой точки поверхности. Таким образом, решение задачи автоматизации проектирования поверхности корпуса судна тесно связано с разработкой автоматических систем программирования, предназначенных для обработки геометрической информации на ЭВМ. Аналогичные задачи имеют место и при подготовке данных для обработки деталей на станках с цифровым программным управлением, когда по типу отрезков и характеру их сочленения требуется вычислить недостающие координаты опорных точек и определить параметры эквидистанты плоского контура.

В настоящее время известен ряд специфических языков, используемых для автоматизации обработки геометрических объектов на ЭВМ. Один из первых языков АРТ (Automatically Programmed Tools) [7, 12] используется при подготовке данных для станков с цифровым программным управлением. Язык АРТ содержит около 300 слов, включающих действия с геометрическими объектами. Об эффективности системы АРТ можно судить по следующему примеру. Подготовка программы работы станка по изготовлению лонжерона крыла вручную занимала 200 ч, с помощью одного из ранних вариантов системы АРТ, содержавшего 107 слов, та же работа выполнялась за 5 ч.

Другой язык COGO [10, 11] предназначен для автоматизации программирования при решении геометрических задач строительной техники. При использовании языка COGO инженер может вводить геометрическую задачу в машину, используя обычный технический язык своей профессии.

Работа системы COGO основана на том, что почти все геометрические задачи строительной механики могут быть сведены к следующему двум элементарным задачам: вычислению координат точки по данной информации и вычислению длины и направления линии между двумя точками.

Управляющая программа COGO в зависимости от условий задачи определяет последовательность указанных элементарных задач и выдает программу решения задачи на языке FORTRAN.

Для автоматизации решения задач начертательной геометрии разработана система Descriptran [9]. Эта система включает библиотеку из 15 стандартных подпрограмм, с помощью которых можно программировать изменения координат данной фигуры в любой из проекций. Среди отечественных работ, связанных с алгоритмизацией обработки геометрических объектов, следует отметить систему автоматизации программирования САП [3, 5], позволяющую готовить информацию для обработки плоских деталей, контуры которых состоят из прямых, окружностей, парабол третьей степени и произвольных плавных кривых, заданных таблично и аппроксимируемых отрезками кубических парабол. Система САП использует вычислительные машины типа М-3 и «Минск-1».

Для вычислительной машины типа «Урал» разработана управляющая программа [1] для обработки плоских контуров, состоящих из отрезков прямых, кусков кривых второго порядка и многочленов n -й степени и участков, заданных таблицей.

Работа С. А. Фролова [6] основана на том, что смысл любого графического построения заключается в нахождении в определенной последовательности точек пересечения двух линий. В связи с этим предлагается 8 элементарных графических операций, с помощью которых можно осуществить решение практически любой задачи, решаемой графическим методом.

Алгоритмизации решения некоторых задач поверхности посвящена работа В. С. Полозова [2].

В настоящей статье рассматриваются общие принципы построения системы автоматизации программирования, предназначенной для обработки геометрической информации. На основе этих принципов автором разработаны интерпретирующие системы, позволяющие автоматизировать составление плазовых книг и развертку листов обшивки судовой поверхности. С помощью этих систем были произведены расчеты различных сечений по ряду проектов. В частности, были составлены плазовые книги для надводных судов следующих типов: земснаряд, нефтерудовоз, цементовоз и самоходная баржа. Для нефтерудовоза была осуществлена развертка листов обшивки носовой и кормовой оконечностей.

Подготовка исходной информации для расчета продольных и поперечных сечений надводного судна в вычислительном центре выполняется одним человеком и занимает около 10—12 ч. Машинное время, которое затрачивается на собственно счет сечений, включая ввод исходной информации и печать результатов, составляет около 40 мин. Построение развертки одного листа в среднем занимает 10—15 сек машинного времени.

Разработанные автором универсальные интерпретирующие системы могут быть использованы для автоматизации различных методов проектирования сложных криволинейных поверхностей. Покажем применение этой методики для расчета поверхностей, спроектированных радиусографическим методом, предложенным Д. С. Китаиновым.

Задача автоматизации составления плазовых книг. При радиусографическом проектировании поверхность задается с помощью нескольких центровых линий, пронизывающих обычно в продольном направлении тело, ограниченное поверхностью, и нескольких продольных линий, лежащих на поверхности. Совокупность этих линий образует радиусографический ключ, с помощью которого определяются координаты центров окружностей, образующих любое поперечное сечение, величины радиусов некоторых из этих окружностей и некоторые координаты опорных точек

сечения. Дальнейшее вычисление недостающих параметров отрезков, образующих сечение, ведется в соответствии с характером сочленения этих отрезков по известным формулам геометрии и тригонометрии. Расчет любых других сечений поверхности (продольных, косых и т. п.) базируется на полном знании аналитических уравнений отрезков, составляющих любое поперечное сечение.

Поперечные сечения поверхности на некотором протяжении качественно сохраняют свою форму (имеется в виду число секторных участков и характер их сопряжения). Расчет сечений такого участка выполняется по одному и тому же алгоритму для каждого значения текущей координаты $x \in [x_1, x_2]$ (значения x_1 и x_2 определяют участок поверхности, построенный по данному закону). Однако на участке с заданным законом построения поверхности каждая линия ключа может быть составлена из нескольких отрезков окружностей и прямых. Поэтому в зависимости от текущего значения координаты x приходится выбирать нужный участок на каждой линии ключа.

Полная информация о поперечных сечениях позволяет получить любые другие сечения. Например, вычисляя при каждом x координаты точек поперечного сечения для заданного набора z , мы получаем координаты точек ватерлиний.

Данные, получаемые при расчете сечений судовой поверхности, могут быть использованы при различных расчетах по статике судна, при развертке листов обшивки и т. п.

Задача автоматизации развертки листов обшивки судовой поверхности. Для автоматизации развертки листов использован метод лучевой развертки, который предполагает, что на проекции «корпус» (аналогичные построения можно выполнить в любой другой проекции) известны аналитические уравнения для секторных участков, составляющих шпангоуты, пазы и стыки листов. Наиболее громоздкую информацию, естественно, составляют данные о шпангоутах, но вся эта информация в нужном виде может быть получена при расчете поперечных сечений судовой поверхности. К шпангоутам формально могут быть причислены и стыки. Это делается для удобства определения местоположения листа на проекции: лист лежит в полосе между шпангоутами N_1, N_2 ($N_1 < N_2$).

Развертка листа лучевым методом осуществляется в два этапа. На первом этапе к листу в некоторой точке C_0 , называемой опорной, проводится касательная плоскость. В действующем варианте программы предполагается, что опорная точка, определяемая проектировщиком, лежит на шпангоуте n и известна одна из ее координат y_n или z_n . Задание только одной из координат опорной точки исключает ошибки проектировщика в определении всех трех координат, которые могут быть допущены особенно на очень крутых или очень пологих участках шпангоутов.

Вообще говоря, наиболее удачной опорной точкой является такая, для которой проекция листа на касательную плоскость имеет максимальную площадь. Вместо касательной плоскости в опорной точке проводится опорная плоскость, достаточно близкая к касательной. Для этого через опорную точку проводятся две прямые: одна является касательной к шпангоуту на плоскости (y, z) , другая лежит достаточно близко к перпендикулярной касательной. След этой перпендикулярной касательной в плоскости (y, z) определяет две точки A_{n-1} и A_{n+1} на шпангоутах, окаймляющих опорный. Вместо перпендикулярной касательной через опорную точку проводится прямая (в пространстве), параллельная хорде A_{n-1}, A_{n+1} . Полученные две прямые и определяют опорную плоскость.

Второй этап — собственно развертка листа. Она производится следующим образом. В пространстве определяются точки, лежащие на контуре листа. Для этого верхний и нижний пазы последовательно пересекаются со шпангоутами. Перебор шпангоутов ведется слева направо, т. е. от N_1 к N_2 . Точку пересечения i -го шпангоута с верхним пазом обозначим через A_{1i} , точку пересечения с нижним пазом — через A_{2i} . Полученные точки проектируются на опорную плоскость, проекции их обозначим через A'_{1i} и A'_{2i} соответственно. Затем определяются длины хорд (первый вариант) или дуг (второй вариант) C_0A_{1i} и C_0A_{2i} .

$$\rho_1 = \delta(C_0, A_{1i}); \quad \rho_2 = \delta(C_0, A_{2i}).$$

На лучах $C_0A'_{1i}$ и $C_0A'_{2i}$ делаются засечки от точки C_0 длиной ρ_1 и ρ_2 соответственно. Полученные точки A''_{1i} и A''_{2i} на опорной плоскости образуют контур развернутого листа. Кроме точек пересечения пазов и шпангоутов, аналогичным образом растягиваются по одной (или более) точки левого и правого стыков.

Естественно, что выдача пространственных координат точек контура развернутого листа неудобна для дальнейшей работы. Поэтому на опорной плоскости некоторым образом вводится система прямоугольных координат (u, v) с началом в точке C_0 , и все пространственные координаты контурных точек развертки переводятся в координаты плоскости (u, v) .

Матричное программирование. Решение поставленных задач связано с вычислениями по сравнительно простым формулам. Однако объем этой несложной по своему характеру работы очень велик из-за большого числа сечений, необходимых на практике. Так, например, объем работы по расчету сечений только корпуса надводного судна составляет около 150—200 тыс. арифметических операций, куда входит значительное число таких трудоемких операций, как извлечение квадратного корня.

Использование электронных вычислительных машин для решения этих задач связано с трудностями программирования процесса перевода алгоритма решения задачи на язык машины. Элементарность и микроскопичность машинных операций приводят к тому, что программы для решения достаточно сложных задач насчитывают несколько тысяч и даже десятков тысяч команд. Для автоматизации этой работы необходимо иметь язык, позволяющий достаточно просто описать исходную информацию и алгоритм решения задачи, и управляющую программу для перевода записи алгоритма с входного языка на язык машины.

На наш взгляд, для решения задач обработки геометрической информации очень удобно использовать запись алгоритма в виде некоторой матрицы P , состоящей из трех частей:

$$P = \{ \{ (A) (R) (B) \} \},$$

где A — прямоугольная матрица размерности $m \times n$;

R — вектор-столбец размерности m ;

B — прямоугольная матрица размерности $m \times l$.

Величины m , n и l определяются особенностями каждого класса задач. Элемент матрицы A можно трактовать как набор аргументов (исходных данных и промежуточных результатов вычислений) или сведений об их расположении в таблицах, информационных списках или в памяти машины; элементы вектор-столбца R — как обобщенные коды операций, выполняемых над элементами соответствующих строк матрицы A ; элементы матрицы B — как сведения о дальнейшем использовании результатов вычислений. В частности, элементами матрицы B могут служить тройки (i, j, k) , определяющие индекс результата вычислений (при операции над элементами одной строки может быть получено несколько величин и, чтобы их различать, им должны быть даны метки — индексы).

В предлагаемой форме записи может быть представлена программа для любой вычислительной машины. Например, если положить $n = 2$, $l = 1$, в качестве матрицы A использовать набор адресов операндов (A_1 и A_2), в качестве R — последовательность кодов операции машины, а в качестве B — набор адресов результатов (A_3), то матрица P превращается в программу для трех-адресной машины. Аналогичным образом можно интерпретировать в матричной форме программы для машин, имеющие другие адресности.

Естественно, что по сравнению с обычной программой такая интерпретация никаких преимуществ не дает из-за элементарности операций машины. Однако для определенных классов задач можно ввести набор обобщенных операций, позволяющих существенно укрупнить этапы вычислений. Например, для решения задач линейной алгебры в качестве таких операций могут

быть взяты сложение, вычитание и умножение матриц, умножение матрицы на скаляр, решение системы линейных алгебраических уравнений и т. п. Элементами частиц A и B в П-программе могут служить сведения о расположении матриц-операндов и их порядках. Сходная по замыслу идея заложена в компиляторе ТМС [8], предназначенном для решения задач линейной алгебры на вычислительной машине IBM-709.

Среди обобщенных команд по аналогии с системой команд универсальной ЦВМ могут содержаться команды условной и безусловной передач управления, организации циклов, печати результатов вычислений и т. п. Вполне понятно, что за счет разумного выбора обобщенных операций и удачной компоновки матриц A и B можно предельно сжать запись алгоритма решения задачи.

Перевод матричных программ в код универсальной ЦВМ может быть выполнен с помощью управляющих систем интерпретирующего или компилирующего типов. В качестве примера использования матричного программирования приведем описание универсальных интерпретирующих систем радиусографического метода (УИС-РГМ и ИС-РГВ), предназначенных для автоматизации расчета сечений и развертки листов обшивки судовых поверхностей на вычислительных машинах типа М-20.

Выбор интерпретации объясняется желанием разместить всю информацию о решаемой задаче в оперативном запоминающем устройстве. Замедление вычислений (в данном случае примерно в 6—8 раз) не вызывает особых возражений, так как вычислительная работа разовая. Кроме того, основное время тратится не на счет, а на печать результатов, время которой в принятой системе интерпретации не увеличивается. При объеме вычислительной работы в 150—200 тыс. арифметических операций и скорости вычислительной машины в 20 000 *оп/сек* такое замедление вполне допустимо.

Набор кодов обобщенных операций в описываемых ниже системах включает всевозможные геометрические построения, выполняемые циркулем и линейкой. Однако этот набор без всякого труда может быть пополнен и другими операциями, позволяющими использовать в построениях не только прямые и окружности, но и любые более сложные кривые (кривые второго порядка, кубические параболы, табличные кривые и т. п.).

Библиотека стандартных подпрограмм. Все графо-аналитические построения, используемые при проектировании поверхности, сведены в библиотеку стандартных подпрограмм БСП-1 или БСП-2, различие между которыми заключается в основном в подходе к заданию прямой. В БСП-1 основной формой задания прямой является уравнение вида

$$y = kx + b.$$

В БСП-2 использован векторный подход к анализу геометрических построений, и прямая считается заданной, если известны две ее точки. УИС-РГМ использует БСП-1 и ИС-РГВ — БСП-2. В БСП-2 включена и стандартная подпрограмма развертки листа.

Почти все стандартные подпрограммы (задачи) построены таким образом, что исходной информацией к ним являются наборы чисел (не более восьми), располагаемые в строго определенной последовательности. Результатом работы стандартных подпрограмм являются числа (от одного до трех), также располагаемые в определенной последовательности (см. табл. 1).

Таблица 1

$N_{сл}$	Исходная информация								Результаты		
	0	1	2	3	4	5	6	7	4	2	1
02	x_1	y_1	x_2	y_2	—	—	—	—	—	b	k
06	k_1	b_1	k_2	b_2	—	—	—	—	—	y	x
70	k	b	—	x_2	y_2	R_2	± 1	—	—	y	x

Здесь задача 02 предназначена для определения уравнения прямой, проходящей через две заданные точки, задачи 06 — для определения точки пересечения двух прямых; задача 70 — для определения точки пересечения прямой с окружностью.

БСП-1 включает 70, а БСП-2 — 45 стандартных подпрограмм, реализующих различные графические, арифметические и вспомогательные операции. По своей структуре библиотека стандартных подпрограмм может быть разбита на следующие группы.

1. Определение координат по данной информации.

Эта группа подпрограмм позволяет определять координаты точки прямой или окружности при различных графических построениях. Например, определение второй координаты точки прямой или окружности по одной из заданных координат; определение точки пересечения двух прямых, двух окружностей, прямой и окружности; определение точки касания двух окружностей; определение точки касания на касательной, проведенной из данной точки к окружности и т. д.

В БСП-1 эта группа состоит из 15 подпрограмм, в БСП-2 — из 16 подпрограмм.

2. Определение недостающих параметров в уравнениях прямой и окружности.

Стандартные подпрограммы этой группы реализуют вычисления по сведению к основной форме представления прямых и окружностей при задании их другими способами. Например, прямая может быть задана точкой и угловым коэффициентом,

двумя точками, точкой и перпендикулярной прямой, параллельной прямой и расстоянием от нее и т. д.

Окружность может быть задана центром и касательной, двумя касательными и радиусом, сопрягаемыми окружностями, сопрягаемыми прямой и окружностью, тремя точками на окружности и т. д. В БСП-1 во вторую группу входит 21 стандартная подпрограмма, в БСП-2 — 16 подпрограмм.

3. Арифметические и логические операции.

Стандартные подпрограммы третьей группы выполняют все арифметические операции, определяют тригонометрические функции узлов, определяют расстояние между точками и проверяют принадлежность точки данному интервалу. В БСП-1 эта группа включает 13 подпрограмм, в БСП-2 — 7 подпрограмм.

4. Вспомогательные операции.

Стандартные подпрограммы четвертой группы используются для организации циклических вычислений и специальных выдач информации (перфорация массива «шпангоут», печать массива «приращения»). Группа состоит из 6 подпрограмм.

5. Прочие операции.

Пятая группа, состоящая из 11 подпрограмм и включенная только в БСП-1, предназначена для расчета сечений линейчатых поверхностей.

Исходная информация для расчета сечений поверхности.

Задание на расчет сечений поверхности оформляется в виде матрицы, содержащей сведения об известных величинах и характере и последовательности графо-аналитических построений для определения недостающих параметров линий или сечений поверхности. Каждому графо-аналитическому построению соответствует определенная стандартная подпрограмма (или некоторая последовательность этих подпрограмм) из библиотеки. В соответствующем столбце матрицы-задания указывается номер нужной задачи. Каждая строка матрицы-задания включает в себя известную числовую информацию, располагаемую в левой части матрицы, номер используемой подпрограммы, сведения о результатах, выдаваемых на печать, и указания о дальнейшем их использовании. Строки матрицы-задания и столбцы левой части ее нумеруются в восьмиричной системе. Порядок нумерации строк может быть произвольным. С помощью введенной нумерации указание, например, об элементе из 5-го столбца 27-й строки матрицы может быть записано в виде 5/27, 27/5 или просто 275.

В результате вычислений по большинству из стандартных подпрограмм получается до трех величин, которым условно присвоены индексы 4, 2, 1. В соответствии с этим графы матрицы-задания, соответствующие указаниям о выдаче результатов на печать и дальнейшем их использовании, разбиты соответственно

на три столбца, порядок заполнения которых определяется решаемой задачей.

Приведем пример матричной программы по определению параметров радиусографической линии, представленной на рис. 1.

Пусть известны координаты y и z точек $O_1 (20; -10)$; $O_2 (22; 30)$; $O_3 (12; 49)$; $A (15; 0)$; $E (60; 72)$. Требуется определить величины радиусов R_1, R_2, R_3 , координаты точек сопряжения B, C, D и параметры уравнения прямой DE .

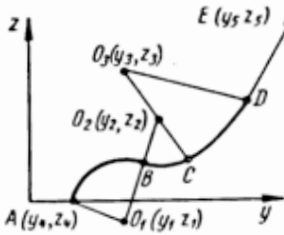


Рис. 1. Пример радиусографической кривой

Последовательность вычислений в данном примере может быть выбрана следующим образом:

а) определяем величину R_1 (задача 12 из БСП-1)

$$R_1 = \sqrt{(y_1 - y_4)^2 + (z_1 - z_4)^2};$$

б) определяем величину R_2 (задачи 12 и 41 из БСП-1)

$$R_2 = \sqrt{(y_1 - y_2)^2 + (z_1 - z_2)^2} - R_1 = n_1 - R_1;$$

в) определяем величину R_3 (задачи 12 и 40 из БСП-1)

$$R_3 = \sqrt{(y_2 - y_3)^2 + (z_2 - z_3)^2} + R_2 = n_2 + R_2;$$

г) из точки E проводим касательную и определяем координаты точки D (задача 16 из БСП-1);

д) по двум точкам E и D определяем параметры прямой (задача 2 из БСП-1);

е) пересечением окружности (R_2, y_2, z_2) с прямыми O_1O_2 и O_2O_3 находим точки сопряжения B и C (задача 70).

Приведенной последовательности вычисления недостающих параметров соответствует определенная матрица-задание (см. табл. 2).

При расчете поперечных сечений участка поверхности, построенного по одному закону, матрица-задание состоит из двух частей.

Первая часть описывает вычисления по ключу, т. е. служит для определения координат некоторых центров окружностей и опорных точек сечения (в приведенном примере эти величины считались известными). Выбор нужного участка линий ключа в зависимости от текущего значения координаты x делается следующим образом. Помимо данных о каждом отрезке составной линии (k, b — для прямых, координат центра и радиуса — для окружности) задаются еще и границы отрезка по одной из переменных. Обработке каждого такого отрезка в матрице-зада-

нии соответствует строка, вычисления по которой следует выполнять только в тех случаях, когда значение текущей координаты попадает в заданные пределы. В противном случае такая строка пропускается.

Таблица 2

№ строк	Исходные данные							$N_{сн}$	Выдача на печать			Дальнейшая адресовка			
	0	1	2	3	4	5	6		7	4	2	1	4	2	1
1	20	-10	15	0					12			R_1			31,
2	20	-10	22	30					12						125
3	n_1	R_1							41			n_1			30
												R_2			51,
4	22	30	12	49					12						135
5	n_2	R_2							40			n_2			50
6	60	72	-	12	49	R_3			16	z_D	y_D		73	72	65
7	60	72	y_D	z_D					02	b	k				
10	20	-10	22	30					02	b_1	k_1		121	120	
11	12	49	22	30					02	b_2	k_2		131	130	
12	k_1	b_1	-	20	-10	R_1	-1		70	z_B	y_B				
13	k_2	b_2	-	22	30	R_2	+1		70	z_C	y_C				

Вторая часть матрицы по найденным значениям центров и точек и по характеру сопряжения отрезков поперечного сечения определяет последовательность вычислений для нахождения значений недостающих параметров, т. е. по своему характеру не отличается от приведенного примера. Полная информация о поперечных сечениях позволяет получать любые другие сечения.

Кодировка исходной информации для расчета сечений. Под числовую информацию каждой строки матрицы по расчету сечений в оперативной памяти машины МОЗУ отводится 10 ячеек¹. Для удобства работы номера ячеек однозначно определяют местоположение элемента в матрице: три старшие цифры адреса указывают номер строки, а младшая — номер столбца, соответствующие данному элементу. Так, например, в ячейке 0325 будет размещен элемент из 5-го столбца 32-й строки. Для сокращения времени подготовки исходной числовой информации в машину вводятся только заданные в матрице ненулевые численные значения, и перед каждой группой ненулевых чисел на бланке пишется адресный код (КА), соответствующий номеру ячейки первого числа группы. К исходной числовой информации относятся и таблицы значений текущих координат x , y , z , для которых в опе-

¹ Здесь и в дальнейшем все величины, связанные с числом ячеек в памяти и их номерами, даются в восьмиричной системе.

ративном запоминающем устройстве отведено фиксированное место. В дальнейшем изложении текущие координаты обозначены через x_1, x_2, x_3 , а таблицы их значений через T_1, T_2, T_3 .

Информация о правой части матрицы, содержащей указания о характере вычислений и порядке использования результатов, располагается в МОЗУ и условно называется псевдопрограммой.

Псевдопрограмма состоит из чередующихся псевдокоманд и строк засылки. Каждая псевдокоманда соответствует одной строке матрицы и занимает одну ячейку памяти, разряды которой используются следующим образом (см. табл. 3):

Таблица 3

П	КОП	A_1	A_2	A_3

в разрядах первой группы (П) записывается сумма индексов элементов, выдаваемых на печать; в разрядах второй группы (КОП) записывается номер $N_{сч}$ используемой задачи; в третьей группе (A_1) записывается номер первой ячейки числовой информации, соответствующей данной строке; в пятой группе (A_3) записывается условное число, доопределяющее характер вычислений по данной строке.

При помощи условных чисел даются указания о выборке очередных значений текущих координат из таблицы в стандартные ячейки, кроме того, условные числа используются для анализа строк матрицы, соответствующих составным линиям. Вслед за псевдокомандой пишутся строки засылок следующим образом: в разрядах П ставится индекс засылаемого результата данной строки матрицы, в разрядах A_3 — номер ячейки, в которую производится засылка.

Для организации циклических вычислений отдельных участков матрицы псевдопрограмма дополняется стандартными строками вида

$$\begin{aligned}
 &0\ 76\ n_1\ l_1\ 0000 \\
 &0\ 74\ n_2\ l_2\ 0000 \\
 &0\ 72\ n_3\ l_3\ 0000
 \end{aligned}
 \tag{1}$$

Строка с кодом 76 используется для организации цикла по координате x_1 , строка с кодом 74 — для организации цикла по координате x_2 , строка с кодом 72 — для организации цикла по координате x_3 . Величины n_1, n_2, n_3 указывают восьмиричную длину таблиц T_1, T_2, T_3 соответственно. Величины l_1, l_2, l_3 указывают номер первой строки циклического участка матрицы по соответ-

ствующей координате. Стандартные строки описанного вида помещаются в псевдопрограмме сразу вслед за описанием последней строки циклического участка матрицы по соответствующей координате. При подготовке исходной информации о матрице сначала пишется псевдопрограмма и вслед за ней — числовая информация. В качестве примера приведем псевдопрограмму для матрицы, представленной в табл. 2 (см. табл. 4).

Таблица 4

1	12	0010		R_1
1			0031	
1			0125	
1	12	0020		n_1
1			0030	
1	41	0030		R_2
1			0051	
1			0135	
1	12	0040		n_2
1			0050	
1	40	0050		R_3
1			0065	
3	16	0060		z_D, y_D
2			0073	
1			0072	
3	02	0070		b, k
3	02	0100		b_1, k_1
2			0121	
1			0120	
3	02	0110		b_2, k_2
2			0131	
1			0130	
3	70	0120		z_B, y_B
3	70	0130		z_C, y_C

Таким образом, кодирование исходной информации для расчета сечений сводится к простому переписыванию данных из матрицы на бланки для программ и не требует никаких специальных знаний, а поэтому может выполняться неквалифицированным персоналом. Время кодирования одной матрицы, содержащей 50—60 строк, составляет 30—45 мин.

Организация и работа интерпретирующей системы УИС-РГМ. Обработка псевдопрограммы по имеющейся числовой информации производится специальной управляющей программой, включающей в себя программу интерпретации и библиотеку стандартных подпрограмм, реализующих графо-аналитические построения. В функции программы интерпретации входят выборка и анализ очередной псевдокоманды, загрузка соответствующей числовой информации в стандартные ячейки, обращение к нужной

стандартной подпрограмме и обработка полученных результатов. Блок-схема программы интерпретации приведена на рис. 2.

Для перебора строк псевдопрограммы используется стандартная ячейка, условно названная «КРА» по аналогии с командным регистром адреса машин. В ней хранится порядковый номер обрабатываемой строки псевдопрограммы.

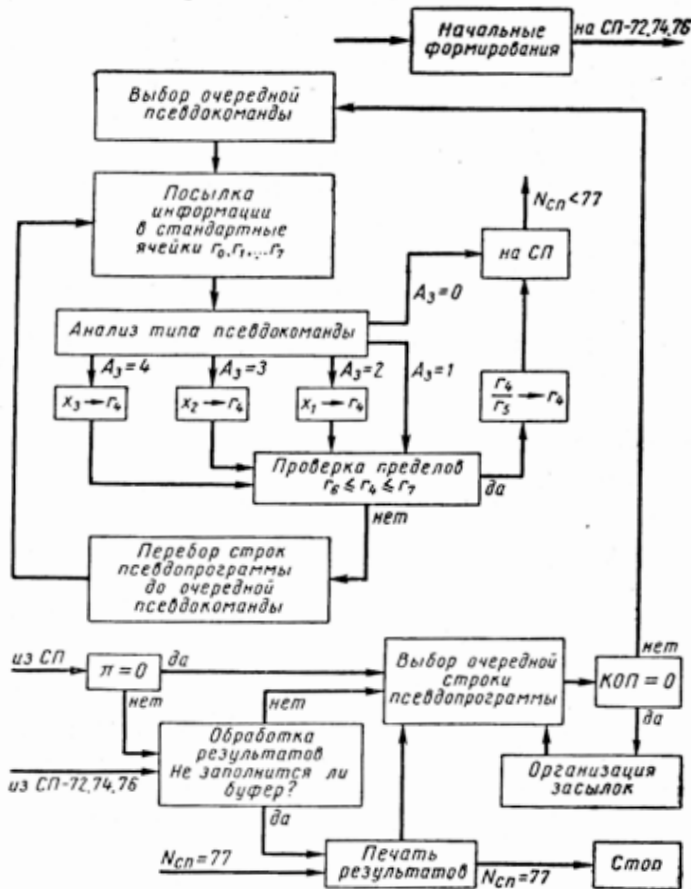


Рис. 2. Блок-схема программы интерпретации

При обработке обычных строк псевдопрограммы «КРА» работает в естественном режиме, т. е. после обработки очередной строки к содержимому «КРА» прибавляется единица. Строки типа (1) нарушают естественный порядок работы псевдопрограммы и выполняют функцию условных передач управления. Обработка этих строк ведется специальной обслуживающей подпро-

граммой. При этом в стандартную ячейку текущей координаты соответствующего цикла засылается очередное табличное значение, определяемое по счетчику, связанному с данной таблицей. К содержимому счетчика каждый раз прибавляется единица до тех пор, пока оно не станет равным длине n_i -й таблицы. Затем организуется поиск в псевдопрограмме строки, у которой старшие три цифры A_1 совпадают с значением l_i . Одновременно с этим в «КРА» фиксируется порядковый номер искомой строки. После окончания цикла соответствующий счетчик гасится и в стандартной ячейке восстанавливается первое табличное значение текущей координаты.

Поиск нужной стандартной подпрограммы (СП) осуществляется с помощью строки таблицы характеристик (ТХ), расположенной в МОЗУ, начиная с ячейки α_0 . Стандартной подпрограмме с номером N соответствует в ТХ строка, расположенная в ячейке $\alpha_0 + N$. Каждая строка ТХ является командой передачи управления на соответствующую стандартную подпрограмму. Все стандартные подпрограммы построены в предположении, что исходная числовая информация расположена в стандартных ячейках $r_0, r_1, r_2, \dots, r_7$. Результаты вычислений по каждой СП помещаются также в стандартные ячейки q_1, q_2, q_3 , соответствующие индексам 4, 2 и 1. Перед результатами счета, выдаваемыми на печать, формируется заголовок строки вида

$$l \quad N,$$

где l равно номеру обрабатываемой строки матрицы, а N — номеру используемой подпрограммы.

Наличие такого заголовка облегчает процесс расшифровки результатов. С целью экономии машинного времени печать результатов производится не после обработки каждой строки матрицы, а после обработки всех строк или в случае заполнения буферного устройства печати.

Признаком конца работы является выборка псевдокоманды с кодом операции 77. Такие строки автоматически записываются на всем рабочем поле псевдопрограммы (перед вводом матрицы).

Автоматизация развертки листов обшивки судовой поверхности. Задача автоматической развертки листов включена в качестве стандартной подпрограммы в БСП-2 системы ИС-РГВ. Информацией для развертки одного листа служат следующие данные:

- а) $y_n(z_n)$ — одна из известных координат опорной точки;
 m — номер листа (в расчете не участвует);
 n — номер опорного шпангоута;
- N_1, N_2 — номера шпангоутов, окаймляющих лист;

- б) описание секторных участков верхнего и нижнего пазов;
- в) таблица описаний секторных участков всех шпангоутов от N_1 до N_2 .

Вся эта информация занимает семь строк матрицы-задания.

Первая строка занята величинами y_n (z_n), m , n , N_1 , N_2 , шесть оставшихся строк отводится под описание секторных участков верхнего и нижнего пазов данного листа (по три строки на каждый паз). Информация о каждом секторном участке занимает 10 ячеек, заполняемых для отрезков прямых и дуг окружностей соответственно в виде таблицы (см. табл. 5).

Таблица 5

0	1	2	3	4	5	6	7
—	y_1	z_1	y_2	z_2	—	—	—
—	y_1	z_1	y_2	z_2	y_4	z_4	R

В табл. 5 (y_1, z_1) и (y_2, z_2) — границы сектора, (y_4, z_4) — координаты центра, а R — радиус.

Таблица описаний секторных участков шпангоутов заполняется в аналогичной последовательности с добавлением в начале каждого сектора (нулевой столбец) величины x (дистанции шпангоута) и размещается либо в МОЗУ, либо на магнитных барабанах в зависимости от длины таблицы.

Числовая информация по развертке листов вводится в машину обычным образом. Псевдопрограмма для развертки в зависимости от места хранения таблицы шпангоутов состоит из строк вида

$$\begin{array}{l} \text{или } 0\ 70\ A_u\ 0000\ 0000 \\ \quad \quad \quad 0\ 75\ A_u\ 0000\ 0000 \end{array}$$

где A_u — адрес, определяющий местоположение числовой информации первой из семи строк матрицы, относящейся к данному листу.

Работа программы начинается с определения коэффициентов нормального уравнения опорной плоскости. Затем определяется средняя точка C_1 на левом стыке листа и ее проекция $C'_1(x', y', z')$ на опорную плоскость. С помощью двух точек C'_1 и C'_0 на опорной плоскости вводится плоская система координат (u, v) , с помощью которой пространственные точки A''_{1i} и A''_{2i} контура развернутого листа снабжаются плоскими координатами. За начало координат выбирается опорная точка C_0 , за ось u — прямая $C'_1C'_0$. Для перевода пространственных координат проекций точек контура на опорной плоскости вводятся базисные точки B_1 и B_2 , у которых

известны пространственные координаты (x, y, z) и их плоскостные координаты (u, v) . Предварительно определяются расстояния $\rho_1(B_1, A')$ и $\rho_2(B_2, A')$, затем из точек B_1 и B_2 делаются засечки радиусами ρ_1 и ρ_2 соответственно. Расстояния ρ_1 и ρ_2 вычисляются по пространственным координатам, а засечки ведутся на плоскости (u, v) по одной из стандартных подпрограмм. Затем точка A' растягивается по лучу C_0A' на длину дуги C_0A . Это построение выполняется на плоскости и, таким образом, точки контура развернутого листа снабжаются плоскостными координатами (u, v) . В качестве длины дуги C_0A принимается длина дуги окружности, проведенной в пространстве через точку A и касающейся опорной плоскости в точке C_0 .

На основании двухлетнего опыта использования разработанных интерпретирующих систем в ряде организаций можно сделать следующие выводы.

Применение систем автоматизации программирования обработки геометрической информации позволяет эффективно использовать вычислительную технику для внедрения прогрессивных методов проектирования судовых поверхностей, основанных на аналитическом, графическом или графо-аналитическом согласовании обводов без применения плаза, что дает возможность не только совершенствовать методы проектирования, но при изменении технологии сократить сроки постройки судов.

Матричное программирование существенным образом уменьшает объем входной информации при автоматизации проектных и конструкторских работ, связанных с обработкой геометрической информации. Изменение набора стандартных подпрограмм делает матричное программирование очень гибким и позволяет легко приспособить описанные интерпретирующие системы для других методов аналитического проектирования сложных криволинейных поверхностей.

Наличие программ интерпретации полностью исключает процесс программирования в коде реальной машины, сводит весь процесс подготовки исходной информации к простой переписи матричных программ на специальных бланках, выполняемой неквалифицированным персоналом.

ЛИТЕРАТУРА

1. Быховский М. Л. и Курочкина А. И. Методы автоматической реализации на математических машинах управляющих программ для обработки плоских контуров. «Труды 3-го совещания по основным проблемам теории машин и механизмов. Теория машин-автоматов и гидродневопривода», М., Машгиз, 1963, стр. 31—57.

2. Полозов В. С. Алгоритмизация решения некоторых задач поверхностей. В сборнике МАИ «Прикладная геометрия поверхностей» под ред. И. И. Котова. М., 1964.

3. Пруден Ю. И. и Тамм Б. Г. Система автоматического программирования обработки деталей. В сборнике «Методы подготовки информации для станков с программным управлением». Под ред. А. Е. Кобринского и Б. Г. Тамма. Таллин, 1963, стр. 55—70.
4. Рейнов М. Н. и др. Судостроительные расчеты на электронных вычислительных машинах. Л., «Судостроение», 1964.
5. Тамм Б. Г. Система автоматизации программирования. «Автоматика и телемеханика», т. XXII, 1961, № 8.
6. Фролов С. А. Алгоритмы графического решения задач на ЭВМ. Сб. «Вычислительная техника», 1964, № 4, М., «Машиностроение», стр. 137—143.
7. Brown S. A., Drayton C. E., Mittman B. A description of the APT language. «Communs. ACM», 1963, 6, N 11, pp. 649—658.
8. Hornick S. D. IBM 709 tape matrix compiler «Communs. ACM», 1959, 2, N 9, pp. 30—32.
9. Kliphardt Raymond A. Descriptran-automated descriptive geometry. «Communs. ACM», 1963, 6, N 6, pp. 336—339.
10. Miller C. L. «COGO-a computer programming system for civil engineering problems» Cabridge, Mass, Technol. Press, 1961.
11. Miller C. L. Computer programming system for civil engineering problems. «Engineers», 1961, 212, № 5516, pp. 629—630.
12. Ross Douglas T. Automatic Programming—«Aircraft Productions», v. 20, N 5, 1958.

О приближённых методах кусочно-линейной аппроксимации плоских кривых

Министерство высшего и среднего специального образования РСФСР

НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ
ПРИКЛАДНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ
И ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ
ГОРЬКОВСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
ИМ. Н. И. ЛОБАЧЕВСКОГО

УЧЕНЫЕ ЗАПИСКИ

ПРИКЛАДНАЯ МАТЕМАТИКА И КИБЕРНЕТИКА

(Материалы к Всесоюзному межвузовскому симпозиуму
по прикладной математике и кибернетике)

ГОРЬКИЙ, 1967

О ПРИБЛИЖЕННЫХ МЕТОДАХ КУСОЧНО-ЛИНЕЙНОЙ АППРОКСИМАЦИИ
" ПЛОСКИХ КРИВЫХ " .

Ю.Л.Кетков

I. АЛГОРИТМ I

В работе [1] при исследовании оптимальных методов кусочно-линейной аппроксимации плоских кривых был использован алгоритм последовательного определения вершин ломаной. При этом отыскание очередной $(n+1)$ -ой вершины ломаной сводилось к решению нелинейного уравнения

$$|d(h)| = \varepsilon, \quad (I)$$

где $d(h)$ - величина стрелки прогиба очередного звена ломаной.

Разложение функции $d(h)$ в ряд Тейлора в окрестности $h=0$ имеет следующий вид:

$$d(h) = \frac{1}{2} d''(0)h^2 + \frac{1}{6} d'''(0)h^3 + \dots \quad (2)$$

В случае явного задания аппроксимируемой кривой $y = f(x)$ коэффициенты этого разложения

$$d''(0) = - \frac{f''(x_n)}{4\sqrt{1+[f'(x_n)]^2}},$$

$$d'''(0) = - \frac{3\{f'''(x_n)(1+[f'(x_n)]^2) - f'(x_n)[f''(x_n)]^2\}}{8(1+[f'(x_n)]^2)^{3/2}}.$$

В случае параметрического задания аппроксимируемой кривой $x = x(t)$, $y = y(t)$ эти коэффициенты вычисляются по формулам

$$d''(0) = - \frac{x'_n y''_n - x''_n y'_n}{4\sqrt{(x'_n)^2 + (y'_n)^2}},$$

$$d'''(0) = \frac{3\{(x'_n y_n''' - x_n''' y'_n)[(x'_n)^2 + (y'_n)^2] - (x'_n y_n'' - x_n'' y'_n)(x'_n x_n'' + y'_n y_n'')\}}{8[(x'_n)^2 + (y'_n)^2]^{3/2}}$$

Для достаточно малого ε , когда мало h и в разложении (2) можно было пренебречь членами $O(h^3)$, в работе [1] было использовано приближенное решение уравнения (1)

$$h = \pm \sqrt{2\varepsilon / |d''(0)|}. \quad (3)$$

Если же ε не очень мало, то (см. таблицу I) такое решение приводит к чересчур завышенным (заниженным) результатам.

Для получения в этом случае более точного решения уравнения (1) мы можем вместо квадратного уравнения решить кубическое уравнение

$$\left| \frac{1}{2} d''(0)h^2 + \frac{1}{6} d'''(0)h^3 \right| = \varepsilon, \quad (4a)$$

воспользовавшись для этой цели приближенным методом [2].

За счет выбора направления аппроксимации (знака h) мы всегда можем считать, что коэффициенты в уравнении (4a) имеют следующие знаки:

$$ah^2 + bh^3 = \varepsilon \quad (a > 0, b < 0). \quad (4б)$$

Подстановка

$$\text{где } \begin{aligned} x &= \omega h, \\ \omega &= -1/\sqrt[3]{\beta}, \quad \beta = -\varepsilon/b, \end{aligned}$$

приводит уравнение (4) к виду

$$g(x) = x^3 + \alpha_1 x^2 - 1 = 0, \quad (4в)$$

$$\alpha_1 = \omega \alpha, \quad \alpha = a/b.$$

Так как $g(0) = -1$, $g(1) = \alpha_1 > 0$, то в интервале $(0; 1)$ $g(x)$ имеет единственный действительный корень x_1 . В противном случае все три корня x_1, x_2, x_3 должны лежать в интервале $(0; 1)$, но это невозможно, так как

$$x_1 \cdot x_2 \cdot x_3 = 1.$$

УКЛОНЕНИЯ ПРИ КУСОЧНО - ЛИНЕЙНОЙ

	ε	h_1	$d(y + h_1)$	h_3
$y=1$	0,25	-1,92846912	0,232399916	2,41903929
	0,1	-1,26234028	0,0979424067	1,44868627
	0,01	-0,419548035	0,0102315873	0,437956291
	0,001	-0,134909228	0,00103096723	0,136754513
$y=2$	0,25	-2,09976240	0,248903797	2,97763633
	0,1	-1,38522226	0,100382686	1,68928290
	0,01	-0,467047530	0,0102183037	0,496116932
	0,001	-0,151315501	0,00103120327	0,154233812
$y=4$	0,25	-2,63867618	0,260223074	3,75327884
	0,1	-1,74099418	0,102014727	2,12593656
	0,01	-0,587159579	0,0102323523	0,623929393
	0,001	-0,190257562	0,00103133759	0,193948951
$y=8$	0,25	-3,66327318	0,258685629	4,75960823
	0,1	-2,40438658	0,101981814	2,81184440
	0,01	-0,803024585	0,0102587602	0,842935826
	0,001	-0,258864467	0,00103148093	0,262868189

Т а б л и ц а I.

АППРОКСИМАЦИИ ПАРАБОЛЫ $y^2 = 4x$. АЛГОРИТМ I.

$d(y+h_3)$	h	$d(y+h)$	$d(y-h)$
0,245437565	2,11474252	0,194828213	0,279393569
0,0993430946	1,33748061	0,0858459603	0,110300682
0,0102364012	0,422948505	0,00956277330	0,0104011342
0,00103098272	0,133748061	0,00986458106	0,00101318380
0,275596336	2,37841423	0,187838282	0,327652833
0,102580778	1,50424123	0,0831381018	0,119982212
0,0102250033	0,475682846	0,00942393735	0,0106110107
0,00103122482	0,150424123	0,000981376887	0,00101897626
0,283664251	2,99069756	0,191182969	0,348819288
0,103781049	1,89148321	0,0838290998	0,122496717
0,0102374919	0,598139513	0,00943186376	0,0106308156
0,00103135405	0,189148321	0,000981394023	0,00101923214
0,267884413	4,06108637	0,201559900	0,327459987
0,202775342	2,55845654	0,0868277733	0,117681089
0,0102611704	0,812217274	0,00954339106	0,0105009209
0,00103148862	0,256845654	0,000985109484	0,00101533287

- 206 -

Для приближенного определения корня x_1 мы можем свести кубическое уравнение (4а) к квадратному уравнению, выразив x^3 из полинома Чебышева $T_3^*(x)$:

$$x^3 \approx \frac{48x^2 - 18x + 1}{32} = 1,5x^2 - 0,5625x + 0,03125.$$

Погрешность такой замены на интервале $(0;1)$ не превосходит $1/32$, т.е. порядка 3%.

Таким образом, приближенное значение корня x_1 выражается формулой

$$x_1 \approx \frac{0,5625 + \sqrt{0,5625^2 + 4 \cdot 0,96875(1,5 + \alpha_1)}}{2(1,5 + \alpha_1)}.$$

Это дает нам один из корней уравнения (4а)

$$h_1 = x_1 / \omega < 0.$$

После исключения найденного корня h_1 из уравнения (4а) для двух других корней имеем

$$h_2 \approx -\frac{\alpha + h_1}{2} + \sqrt{\left(\frac{\alpha + h_1}{2}\right)^2 - h_1(\alpha + h_1)},$$

$$h_3 \approx -\frac{\alpha + h_1}{2} - \sqrt{\left(\frac{\alpha + h_1}{2}\right)^2 - h_1(\alpha + h_1)}.$$

Нетрудно видеть, что $h_2 > h_3 > 0$.

Изложенный алгоритм был использован для аппроксимации параболы $y^2 = 2px$. При этом поиск очередной вершины ломаной был организован по координате y ($y_{n+1} = y_n + h$). Соответствующие коэффициенты кубического уравнения имеют вид

$$a = \frac{1}{8\sqrt{p^2 + y^2}},$$

$$b = -\frac{y}{16\sqrt{(p^2 + y^2)^3}},$$

$$\alpha = -\frac{2(p^2 + y^2)}{y}, \quad \omega = -\frac{1}{\sqrt{p^2 + y^2}} \sqrt[3]{\frac{y}{16\varepsilon}},$$

$$\rho = \frac{16\epsilon\sqrt{(p^2 + y^2)^3}}{y}, \quad \alpha_1 = \sqrt[3]{\frac{p^2 + y^2}{2y^2\epsilon}}.$$

Численный эксперимент (см. таблицу I) показал, что при сравнительно больших ϵ h_1 дает удовлетворительный шаг для аппроксимации назад, а корень h_3 - для аппроксимации вперед. Корень h_2 следует отбросить. Для сравнения в таблице I приведены результаты аппроксимации с шагом h , подсчитанным по формуле (3).

2. АЛГОРИТМ 2

Второй алгоритм использует метод неопределенных коэффициентов для уточнения приближенного решения (3). При этом более точное решение ищется в виде

$$H_k = a_1 \delta + a_2 \delta^2 + \dots + a_k \delta^k \quad (\delta = \sqrt{\epsilon}).$$

Применим этот подход к кусочно-линейной аппроксимации параболы $y^2 = 2px$. Если H - шаг по координате y , то уравнение (I) для параболы имеет вид

$$\frac{H^2}{4\sqrt{4p^2 + (2y \pm H)^2}} = \epsilon = \delta^2. \quad (5)$$

Аппроксимации вперед здесь соответствует знак "+", аппроксимации назад - знак "-".

а) П е р в о е п р и б л и ж е н и е ($k=1$). Подставим первое приближение $H_1 = a_1 \delta$ в уравнение (5):

$$a_1^4 = 64(p^2 + y^2) + 64y a_1 \delta + 16a_1^2 \delta^2. \quad (6)$$

После отбрасывания членов, содержащих множитель $O(\delta)$, получим

$$a_1 = \sqrt[4]{8(p^2 + y^2)}$$

и это дает уже известное [1] решение

$$H_1 = \sqrt{8\epsilon} \sqrt{p^2 + y^2}.$$

б) В т о р о е п р и б л и ж е н и е ($k=2$) для аппроксимации вперед. В [1] было показано, что $d(y+H_1) < \epsilon$, поэтому второе приближение следует искать в виде $H_2 = a_1 \delta + a_2 \delta^2$.

Подстановка его в (5) дает

$$a_1^4 + 4a_1^3 a_2 \delta + 6a_1^2 a_2^2 \delta^2 + 4a_1 a_2^3 \delta^3 + a_2^4 \delta^4 = 64(p^2 + y^2) + 64y a_1 \delta + 16(a_1^2 + 4y a_2) \delta^2 + 32 a_1 a_2 \delta^3 + 16 a_2^2 \delta^4.$$

После отбрасывания членов, содержащих множитель $O(\delta^2)$,

$$a_1^4 + 4a_1^3 a_2 \delta = 64(p^2 + y^2) + 64y a_1 \delta.$$

Отсюда

$$a_1 = \sqrt{8 \sqrt{p^2 + y^2}}, \quad a_2 = 2y / \sqrt{p^2 + y^2}.$$

Таким образом, получено второе приближение

$$H_2 = \sqrt{8\epsilon \sqrt{p^2 + y^2}} + \frac{2y\epsilon}{\sqrt{p^2 + y^2}}.$$

Можно показать, что при кусочно-линейной аппроксимации с шагом

H_2 для любого ϵ справедливо неравенство

$$d(y + H_2) < \epsilon.$$

в) Т р е т ь е п р и б л и ж е н и е ($k=3$) для аппроксимации вперед. Аналогичным образом определяется третье приближение:

$$H_3 = \sqrt{8\epsilon \sqrt{p^2 + y^2}} + \frac{2y\epsilon}{\sqrt{p^2 + y^2}} + \frac{(2p^2 + y^2)\epsilon \sqrt{\epsilon}}{\sqrt{2(p^2 + y^2)^5}}.$$

Численный пример (см. таблицу 2) показал, что при аппроксимации вперед с шагом H_3 могут получиться звенья, уклоняющиеся от параболы более чем на ϵ .

г) В т о р о е п р и б л и ж е н и е ($k=2$) для аппроксимации назад. Для аппроксимации назад известно [1], что шаг H_1 слишком велик (он почти всегда выводит ломаную из ленты ширины ϵ). Поэтому второе приближение имеет смысл искать в виде

$$\bar{H}_2 = a_1 \delta - a_2 \delta^2.$$

Таблица 2

КУСОЧНО-ЛИНЕЙНАЯ АППРОКСИМАЦИЯ ПАРАБОЛЫ $y^2=4x$, Алгоритм 2

$$1 \leq y \leq 32$$

ε	АППРОКСИМАЦИЯ ВПЕРЕД			АППРОКСИМАЦИЯ НАЗАД		
	$\frac{r_1}{\Delta}$	Минимальное уклонение	Максимальное уклонение	$\frac{r_2}{\Delta}$	Минимальное уклонение	Максимальное уклонение
0,25	H_2	0,231652214	0,248775023	\bar{H}_2	0,240045976	0,248892140
	H_3	0,249881568	0,250052810	\bar{H}_3	0,250001084	0,250059759
0,1	H_2	0,0966403036	0,0998301268	\bar{H}_2	0,0973711959	0,0998306267
	H_3	0,0999873931	0,100004569	\bar{H}_3	0,1000000066	0,100003764
0,01	H_2	0,00996162175	0,0099834602	\bar{H}_2	0,00996815272	0,0099839497
	H_3	0,0099998016	0,0100000017	\bar{H}_3	0,0099999634	0,0100000045
0,001	H_2	0,000999603073	0,00099984006	\bar{H}_2	0,000999629863	0,00099984221
	H_3	0,00099999978	0,00100000000	\bar{H}_3	0,00099999986	0,00100000000

- 210 -

В этом случае получены следующие результаты:

$$\bar{H}_2 = \sqrt{8\epsilon \sqrt{p^2 + y^2}} - \frac{2y\epsilon}{\sqrt{p^2 + y^2}}, \quad d(y - \bar{H}_2) < \epsilon.$$

д) Третье приближение ($k=3$) для аппроксимации назад. Так как аппроксимация назад с шагом \bar{H}_2 еще не выводит из ленты ширины ϵ , то третье приближение имеет смысл искать в виде

$$\bar{H}_3 = a_1 \delta - a_2 \delta^2 + a_3 \delta^3.$$

Для \bar{H}_3 получено следующее выражение:

$$\bar{H}_3 = \sqrt{8\epsilon \sqrt{p^2 + y^2}} - \frac{2y\epsilon}{\sqrt{p^2 + y^2}} + \frac{(2p^2 + y^2)\epsilon\sqrt{\epsilon}}{\sqrt{2(p^2 + y^2)^5}}.$$

Численный пример (см. таблицу 2) показывает, что при аппроксимации назад с шагом \bar{H}_3 могут получаться звенья, уклоняющиеся от параболы более чем на ϵ .

3. КАЧЕСТВО ПРИБЛИЖЕННЫХ МЕТОДОВ КУСОЧНО-ЛИНЕЙНОЙ АППРОКСИМАЦИИ

Качество различных алгоритмов кусочно-линейной аппроксимации плоских кривых можно характеризовать неравномерностью полосы аппроксимации. Если на расстоянии ϵ от заданной кривой провести эквидистанту, то каждое звено оптимальной вписанной ломаной должно касаться эквидистанты, а вся ломаная будет лежать в получившейся полосе ширины ϵ .

Различные приближенные алгоритмы приводят к ломаным, звенья которых могут как целиком лежать внутри оптимальной полосы, так и выходить за ее пределы. Получающаяся при этом полоса, содержащая ломаную, будет отличаться от оптимальной. Назовем коэффициентом неравномерности полосы аппроксимации величину

$$\lambda = \max_i \frac{|\varepsilon - |d_i||}{\varepsilon} 100,$$

d_i - стрелка прогиба i -го звена ломаной. По данным таблицы 1 и 2 может быть составлена следующая таблица.

Т а б л и ц а 3

Неравномерность полосы при аппроксимации параболы $y^2 = 4x$ на участке $1 \leq y \leq 32$

ε	λ							
	Метод [1]		Алгоритм 1		Алгоритм 2, $k=2$		Алгоритм 2, $k=3$	
	впе- ред	на- зад	впе- ред	на- зад	впе- ред	на- зад	вперед	назад
0,25	24%	40%	12%	8%	8%	4%	0,04 %	0,02%
0,1	17%	22%	4%	2%	3%	3%	0,01 %	0,004%
0,01	6%	6,5%	2,5%	2,6%	0,4%	0,3%	0,0002%	0,00004%
0,001	2%	2%	3%	3%	0,04%	0,04%	0,000002%	0,000001%

Приведенная таблица свидетельствует о достаточно высокой эффективности предлагаемых методов кусочно-линейной аппроксимации плоских кривых.

Л И Т Е Р А Т У Р А

1. Ю.Л.Кетков, Об оптимальных методах кусочно-линейной аппроксимации, Изв. высш.уч.зав.-Радиофизика, 9, №6, 1202 (1966)
2. К.Ланцос, Практические методы прикладного анализа, Физматгиз, М., 1961.

НИЖЕГОРОДСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

На правах рукописи

КЕТКОВ Юлий Лазаревич

**СОЗДАНИЕ ИНСТРУМЕНТАЛЬНЫХ
ПРОГРАММНЫХ СРЕДСТВ ДЛЯ РАЗРАБОТКИ
ДИАЛОГОВЫХ СИСТЕМ
САПР И АСНИ**

05.13.12.– системы автоматизированного проектирования

**Диссертация
на соискание ученой степени
доктора технических наук
(в форме научного доклада)**

Нижний Новгород, 1992

Работа выполнена в Научно-исследовательском институте прикладной математики и кибернетики при Нижегородском ордена Трудового Красного Знамени государственном университете им. Н.И. Лобачевского

Официальные оппоненты

Доктор технических наук, профессор Зайцева Ж.Н.
Доктор технических наук, профессор Дмитриевич Г. Д.
Доктор технических наук, профессор Полозов В.С.

Ведущее предприятие – Научно-исследовательский институт измерительных систем (г. Нижний Новгород)

Защита состоится «___» _____ 1992 г
в ___ часов на заседании специализированного совета Д. 063. 85. 02
при Нижегородском политехническом институте

Заверенные отзывы просим присылать по адресу: 603600, ГСП-41
г. Нижний Новгород, ул. Минина, 24, специализированный совет Д.063.85.02

С диссертацией можно ознакомиться в научно-технической библиотеке
института

Диссертация разослана «_____» _____ 1992 г

Ученый секретарь
Специализированного совета,
Кандидат технических наук

Иванов А.П

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

АКТУАЛЬНОСТЬ ТЕМЫ. История развития средств вычислительной техники и опыт ее использования в сфере автоматизации научно-исследовательских и опытно-конструкторских работ доказали, что применение ЭВМ для решения трудно формализуемых задач наиболее эффективно при диалоговом взаимодействии с человеком. В этом режиме удачно сочетаются производительность ЭВМ с интуицией и опытом исследователя, с его умением оценивать нештатные ситуации, быстро анализировать сложные графические изображения.

В связи с переходом от эксплуатации систем проектирования из пакетного режима в интерактивный возрастает роль работ по созданию программных средств поддержания различных форм диалога в САПР и АСНИ, развития и внедрения диалоговых систем программирования и систем машинной графики, расширения возможностей общения с ЭВМ в рамках традиционных языков программирования и операционных систем.

Эта обширная область научных исследований и практических приложений постоянно привлекает внимание многих отечественных и зарубежных ученых. И если проблематика развития графических компонент автоматизированных систем зачастую упирается в несовершенство доступных нам технических средств, то в технологии создания диалоговых САПР и АСНИ наблюдается серьезный прогресс.

Заметный вклад развитие этого направления внесли В.И. Анисимов, В.М. Брябрин, И.В. Вельбицкий, Г.Д. Дмитриевич, Ж.Н. Зайцева, В.П. Иванов, Н.Г. Малышев, И.П. Норенков, Э.Х. Тыугу и многие другие. К числу крупных достижений в области машинной графики в нашей стране относятся работы Ю.М. Баяковского, В.П. Ершова, Д.Н. Тодороя. Большой интерес на Всесоюзных конференциях вызывают работы в области компьютерной графики, представленные воспитанниками школ В.С. Полозова, В.И. Скурихина, В.Л. Каткова, А.Г. Горелика, О.И. Семенкова и др.

Диссертационная работа выполнена в период 1968-1992 гг. в соответствии с планом госбюджетных НИР и ОКР НИИ прикладной математики и кибернетики по решению Комиссии Президиума СМ СССР по ВПВ N 12 от 25.01.67, Постановлению ГКНТ СМ СССР N 390 от 5.11.76, заданию ГКНТ О.Ц.027.06.16 по целевой комплексной программе по АСНИ и САПР, координационным планам АН СССР по проблеме «Кибернетика» на 1976-1985 гг., решению Комиссии Президиума СМ СССР по ВПВ N 251 от 11.07.83 г., дополнительному заданию N 10 к Постановлению ГКНТ N 263 от 5.06.85 и в рамках хозяйственных договоров с СКБ «Лазурит», ПО Гидромаш, ПМЗ «Восход», Павловским механическим заводом, 280 ЦКП ВМФ, Поволжским лесостроительным предприятием.

ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ. Целью работы является решение крупной научно-технической проблемы, имеющей важное народнохозяйственное значение в части разработки методов построения диалоговых и графических компонент САПР и АСНИ, обеспечивающих более эффективное их использование и направленных на повышение качества изготовления сложноструктурированных графических документов.

В соответствии с целью работы сформулированы следующие задачи исследования:

- изучение возможностей алгоритмических языков, используемых при создании САПР и АСНИ, разработка методов пошаговой компиляции диалоговых программ, построение пакетных и многотерминальных систем программирования;
- расширение диалоговых возможностей языков программирования, традиционно используемых для построения САПР и АСНИ, и развитие интерактивных компонент операционных систем ЕС ЭВМ;
- создание интерактивных графических компонент для подготовки издательских оригиналов в системах автоматизированной обработки картографической информации;
- апробация разработанных методов и систем в конкретных САПР и АСНИ.

НАУЧНАЯ НОВИЗНА

1. В период 1969-1971 гг. разработаны первые отечественные диалоговые системы программирования на базе алгоритмического языка БЕЙСИК, обеспечившие возможность создания интерактивных компонент САПР и АСНИ. В отличие от аналогичных продуктов зарубежного производства созданные системы вместо интерпретации использовали принцип компиляции, повышающий в 30-50 раз быстродействие рабочих программ. За рубежом БЕЙСИК-системы компилирующего типа появились спустя 5-10 лет.

2. С существенным опережением зарубежных разработок в состав языка БЕЙСИК были введены средства подключения автокодных (машинных) команд и форматного вывода числовой информации, что позволило использовать ранее созданные библиотечные проектные процедуры и повысить качество вывода таблично-ориентированных документов САПР и АСНИ.

3. Предложен метод пошаговой компиляции и выполнения диалоговых программ, обеспечивающий многотерминальное обслуживание без тиражирования в оперативной памяти системных программ (компилятор, библиотеки стандартных процедур, модули управления обменом с внешними устройствами), что явилось прототипом вычислительных систем коллективного пользования.

4. В составе операционной системы ДОС ЕС, ориентированной преимущественно на пакетный режим работы, разработан многотерминальный диалоговый редактор символьных документов, не имеющий прототипов и аналогов, что позволило на базе ЭВМ типа ЕС-1035 создать ряд интерактивных автоматизированных систем.

5. На базе дисплейной станции ЕС-7920 разработан пакет программ ДИАФОР, расширяющий возможности ФОРТРАНА ЕС и обеспечивающий конструирование диалоговых сценариев с «жесткой» и «управляемой» схемами взаимодействия, что позволило реализовать принцип изменения активности сторон в процедурах принятия сложных проектных решений, управлять логической схемой проектирования и вести общение с прикладной программой на языке, близком к естественному.

6. Разработана архитектура диалоговой графической компоненты оформления проектной документации, включенной в состав первой отечественной картографической системы. Предложен графический макроязык для формирования библиотек типовых графических элементов, обеспечивающий прецизионное представление и отображение условных знаков и символов картографических шрифтов.

7. Инструментальные программные средства, разработанные в рамках диссертационной работы, объединены общим функциональным назначением. Они направлены на развитие диалоговых систем программирования, совершенствование интерактивных возможностей операционных систем и традиционных алгоритмических языков и, тем самым, составляют базу для создания диалоговых и графических компонент САПР и АСНИ.

ПРАКТИЧЕСКАЯ ЦЕННОСТЬ РАБОТЫ И РЕЗУЛЬТАТЫ ВНЕДРЕНИЯ

1. В период 1971-1986 гг. различные версии БЕЙСИК-системы были переданы 96 организациям, среди которых 54 отраслевые НИИ, КБ и промышленные предприятия, 21 высшее учебное заведение, 9 академических институтов. На базе этих разработок в ГГУ был создан первый в стране учебный терминал-класс. Проект БЕЙСИК-ЕС демонстрировался в составе материалов экспозиции СССР на международной выставке «Средства ЕС и СМ ЭВМ и их применение» (Москва, 1979).

2. На базе пакета ДИАФОР в рамках САПР ПО Гидромаш подсистема расчета устойчивости авиационных шасси была переведена из пакетного режима в диалоговый.

3. Инструментальные средства пакета ДИАФОР использованы в составе САПР «Алмаз», предназначенной для решения задач оптимального раскроя заготовок сложной формы в системах автоматизированного технологическо-

го проектирования и конструирования, внедренных в ЦНИИ «Буревестник» (г.Горький) и НПО ПАРМА (г.Пермь), САПР «Авангард», внедренной на Дмитровском заводе фрезерных станков.

4. В 1985-1990 гг. в практику работы подразделений Главного Управления Геодезии и Картографии, Центрального картпроизводства ВМФ, Поволжского лесоустроительного предприятия была внедрена серия интерактивных подсистем документирования процесса картсоставления с помощью АСНИ «Аquareль», «Асоимк», «Лес» и др.

За разработку соответствующих инструментальных программных средств автор в 1991 г. в составе творческого коллектива был удостоен премии Совета Министров СССР.

АПРОБАЦИЯ РАБОТЫ. Основное содержание работы докладывалось на 2-й Всесоюзной конференции по программированию (Новосибирск, 1970), 2-й Всесоюзной конференции по применению ЭВМ и математических методов в планировании и управлении предприятиями (Горький, 1971), Всесоюзном симпозиуме «Теория языков и методы построения систем программирования» (Алушта, 1972), 3-м конгрессе математиков БНР (Варна, 1972), Всесоюзных конференциях по математическому обеспечению ЭВМ типа М-20 (Москва, 1973, 1975), Всесоюзной конференции «Математическое обеспечение моделирования сложных систем» (Киев, 1977), Республиканском симпозиуме «Теория и практика системного программирования» (Винница, 1977), 3-й Всесоюзной школе «Математическое обеспечение АСУП» (Горький, 1978), Всесоюзной конференции «Методы математической логики в проблемах ИИ и систематическое программирование» (Паланга, 1980), Всесоюзной конференции «Автоматизация производства ППП» (Таллин, 1980), Всесоюзной конференции «Методы трансляции» (Новосибирск, 1981), 4-й Всесоюзной конференции «Интерактивная технология в САПР» (Таллин, 1981), 3-й Всесоюзной конференции «ДИАЛОГ-83» (Протвино, 1983), 4-й Всесоюзной конференции «Системное и теоретическое программирование» (Кишинев, 1983), 1-й, 2-й и 3-й Всесоюзных конференциях «Методы и средства обработки сложной графической информации» (Горький, 1983, 1985, 1988), Всесоюзной конференции «Методы трансляции и конструирования программ» (Новосибирск, 1988), а также на семинарах кафедры Нижегородского университета и НИИ прикладной математики и кибернетики.

ПУБЛИКАЦИИ. Всего по теме диссертации опубликована 61 работа, среди которых 7 монографий, 8 препринтов и учебных пособий, 12 статей, 29 тезисов докладов, 5 алгоритмов и пакетов программ, зарегистрированных в ГосФАП СССР. В текст доклада включен перечень из 49 работ, в которых нашли отражение основные научные результаты, положения и выводы диссертации.

ционной работы. К этому списку добавлены 5 отчетов по НИР, отражающих наиболее характерные внедрения в отраслевых САПР и АСНИ.

2. ВВЕДЕНИЕ

Наименования инструментальных средств, включенных в состав диссертации, и их место в обобщенной схеме САПР (АСНИ) приведены в левой колонке на рис.1.

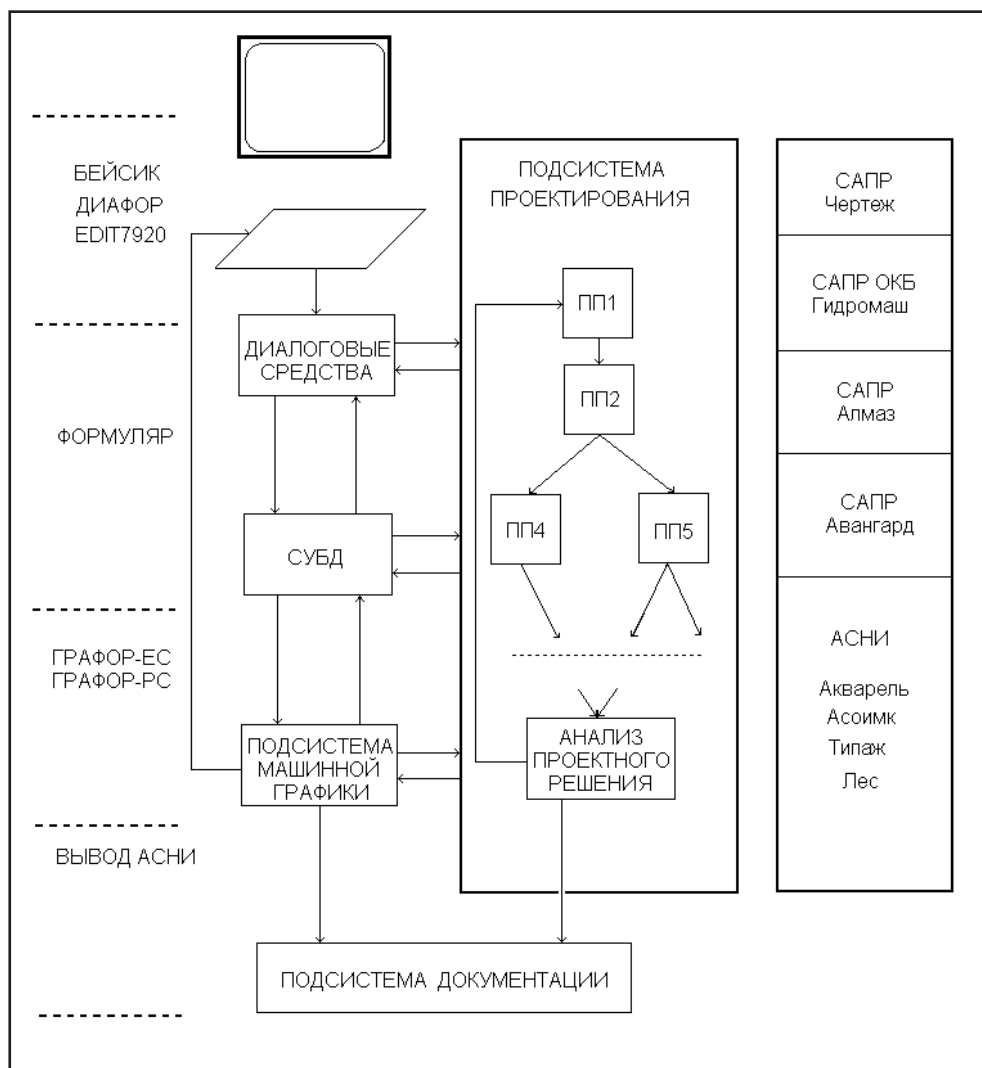


Рис.1.

В правой колонке приведены наименования автоматизированных систем проектирования (САПР) и АСНИ, в рамках которых внедрены разработанные средства. Некоторые из этих приложений выполнены под руководством и при непосредственном участии автора (САПР «Чертеж» - автоматизированная система проектирования изделий судостроительной промышленности, САПР ОКБ Гидромаш - подсистема анализа устойчивости авиационных шасси, автоматизированные картографические системы - АСНИ). В других приложениях - САПР «Алмаз» (система автоматизированного раскроя листового материала, руководитель работ к.ф.-м.н. Л.Б. Белякова), САПР «Авангард» (система автоматизации проектирования консольных вертикально-фрезерных станков, руководитель работ д.т.н. Ю.И. Городецкий) - использованы соответствующие разработки автора.

3. ДИАЛОГОВЫЕ СИСТЕМЫ ПРОГРАММИРОВАНИЯ НА БАЗЕ ЯЗЫКА БЕЙСИК

[2, 3, 7, 8, 9, 13, 14, 15, 16, 17, 19, 20, 23, 32, 36, 46, 48, 49]

Проблема проектирования диалоговой системы программирования была сформулирована в 1968 г., когда автор был прикомандирован в ИПМ АН СССР для участия в комплексных работах по созданию принципиально новой операционной системы БЭСМ-6 [1,11,12].

В качестве входного языка был выбран практически неизвестный в нашей стране БЕЙСИК - один из первых диалоговых алгоритмических языков, созданный в 1964 г. учеными Дартмутского колледжа Дж. Кемени (J. Kemeny) и Т.Куртцем (T.Kurtz). Первоначально БЕЙСИК использовался как средство быстрого приобретения навыков программирования в системах коллективного пользования. С появлением мини-ЭВМ, встраиваемых в комплексы управления лабораторными экспериментами и технологическими процессами, БЕЙСИК стал одним из ведущих языков для разработки АСНИ.

Включение в его состав процедур обработки текстовой и графической информации, появление широкого набора файловых операций и средств взаимодействия с активными устройствами делает возможным его использование для создания автоматизированных систем проектирования. По сравнению с традиционно использовавшимся ФОРТРАНОМ язык БЕЙСИК имеет один очевидный недостаток. В большинстве его версий за исключением последних программных продуктов фирмы Microsoft Corp. (Quick BASIC, Visual BASIC) достаточно слабо представлены средства модульного программирования, что затрудняет построение больших программных систем.

Первые отечественные системы программирования на базе языка БЕЙСИК были разработаны в 1969-1970 гг. под руководством автора для ЭВМ типа

М-20. В отличие от серии BASIC-интерпретаторов, появившихся за рубежом, отечественные системы использовали принцип пошаговой компиляции, что позволило более чем на порядок повысить быстродействие рабочих программ. Наряду с диалоговым режимом общения система БЕЙСИК-222 обеспечивала выполнение задач и в пакетном режиме, превосходя по скорости все штатные системы программирования. Задержка между моментом начала трансляции и выдачей результатов составляла порядка 1-2 сек. Транслирующие системы со столь высокой реактивностью появились лишь 15 лет спустя на персональных ЭВМ в программных продуктах фирм Borland International (Turbo-системы) и Microsoft Corp. (Quick-системы).

В отличие от Бейсик-систем зарубежного происхождения, длительное время эксплуатируемых на СМ ЭВМ, все разработанные нами версии изначально были ориентированы на многотерминальное обслуживание без тиражирования компилятора в памяти. В них был предложен и реализован механизм ускоренного переключения задач пользователей. Для каждого из них выделялась своя область оперативной памяти USAREA, в которой наряду с окном для пользовательской программы хранились и все указатели на соответствующие поля. Поэтому процедура переключения с одной задачи на другую сводилась к перезагрузке регистра базы USAREA и регистров общего назначения.

Разработка диалоговых систем программирования сопровождалась подключением новых оригинальных средств и возможностей к составу операторов входного языка. Отметим среди них наиболее важные. Во-первых, было реализовано управление форматом выдачи числовой информации, что позволило повысить качество вывода таблично-ориентированных документов. С этой целью в состав оператора PRINT была включена функция FMT для установки текущего формата вывода :

100 PRINT FMT(I8), A, B, FMT(F6.2), C

Во-вторых, пользователь получил возможность включить непосредственно в текст БЕЙСИК-программы группу машинных (автокодных) команд :

200 БЭСМ КОП1, A11, A12, A13

210 БЭСМ КОП2, A21, A22, A23

.....
260 БЭСМ

Здесь КОП - числовой код операции, A_{ij} - операнды команд, в качестве которых могли выступать номера строк, абсолютные и относительные адреса, имена переменных и массивов, литеральные константы. Таким образом можно было получить доступ к библиотечным процедурам, например для вывода графических документов. В-третьих, матричные операторы предусматривали возможность ввода/вывода фрагментов массива :

300 MAT PRINT A(5:12), B(1:4,3:8)

Наконец, в версии БЕЙСИК-ЕС был реализован оператор COPY, который позволял включить в свою задачу заранее подготовленные БЕЙСИК-фрагменты с одновременной настройкой их по месту в тексте исходной программы.

В процессе разработки Бейсик-компиляторов был создан широкий спектр инструментальных средств в виде набора макрокоманд на АВТОКОДЕ БМ-4/220 и АССЕМБЛЕРЕ ЕС ЭВМ. Некоторые из них были зарегистрированы в государственном фонде алгоритмов и программ, а впоследствии использовались при разработке новых диалоговых систем.

По решению математического совета Комиссии по эксплуатации ЭВМ при Президиуме АН СССР система БЕЙСИК-222 была рекомендована к включению в состав штатного математического обеспечения ЭВМ типа М-20. На ее базе в ГГУ был создан первый в стране терминал-класс.

В процессе создания системы Бейсик-ЕС была отработана технология взаимодействия с дисплейными станциями ЕС-7920 под управлением ДОС ЕС. Тем самым было ликвидировано очередное белое пятно в штатном математическом обеспечении ЕС ЭВМ и подготовлена почва для создания диалоговых прикладных программ на базе дисплейных станций указанного типа.

Последующие работы автора, связанные с Бейсиком, имели целью анализ новых возможностей языка и исследование путей его развития. Проанализировано большое количество отечественных и зарубежных Бейсик-систем на ЭВМ различных поколений, предложена классификация языковых средств и составлен толковый словарь языка, насчитывающий более 1000 операторов и функций.

4. ТАБЛИЧНЫЙ ИНТЕРФЕЙС МЕЖДУ МОДУЛЯМИ САПР ПО ДАННЫМ

[9, 22, 25]

В 1968-1977 гг. коллектив сотрудников НИИ ПМК, возглавляемый автором, принимал участие в создании системы автоматизированного проектирования изделий судостроительной промышленности САПР «Чертеж». В работе были задействованы ИК АН УССР (головная организация), СКБ «Лазурит», ОКБ БИМК и одна из проектных организаций ВМФ СССР. Задача НИИ ПМК заключалась в реализации большого комплекса алгоритмов научно-исследовательского и эскизного проектирования. Отработка проектных алгоритмов и интерфейса между ними проводилась на БЭСМ-3М, окончательный вариант функционировал на БЭСМ-6.

Среди инструментальных средств этого комплекса наибольший интерес представляет информационно-поисковая система «Формуляр», разработанная в 1973 г. и обеспечивающая интерфейс между проектными модулями. Интерф-

мация, с которой пришлось столкнуться при программировании алгоритмов, оказалась достаточно разнообразной как по своей природе, так и по форме представления. Сюда входили технические данные проекта, паспортные характеристики комплектующих изделий, сведения о прототипах, всевозможные таблицы, эмпирические зависимости и т.п. Для их хранения была выбрана единойобразная табличная форма, названная формуляром.

С точки зрения внешнего представления формуляр представлял собой таблицу, состоящую из последовательности строк переменной длины, в которых наиболее существенными были позиции номера строки, наименования (идентификатора) параметра, значения параметра, формата его представления и информации о режимах использования (получается из расчета, вводится с терминала, разрешены чтение и запись, только чтение или только запись и т.п.). Вместо значения параметра в формуляре могла присутствовать ссылка на другой формуляр. Такая возможность существенно упрощает логику работы модулей проектирования, которым не обязательно «знать», где будут использоваться их выходные результаты. Поэтому они и не обязаны заботиться о тиражировании своих результатов в формулярах всех последующих расчетов, ибо при подключении новых модулей или при изменении последовательности и состава проектных алгоритмов в каждую из существующих программ приходилось бы вносить изменения.

Процедура ввода формуляров обеспечивала построчный прием описания очередной записи. Наличие номеров строк позволяло вести редактирование в процессе ввода. Каждая вновь поступившая строка затирала предыдущую, имевшую такой же номер. Для исключения строки из формуляра достаточно было после номера строки нажать клавишу «Ввод». После ввода описаний все строки упорядочивались по возрастанию их номеров. Этот сравнительно несложный механизм, заимствованный из БЕЙСИКА, был особенно эффективен в режиме диалога. Основной состав формуляров готовился на перфокартах и вводился в автоматическом режиме.

ИПС обеспечивала работу с библиотеками формуляров – разбиение их на группы (разделы) в соответствии с технологией проектирования, редактирование, перемещение наиболее часто используемых формуляров во временную библиотеку с меньшим временем доступа, распечатку каталогов и формуляров. В ее составе были предусмотрены процедуры обмена с возможностью поиска параметра как по номеру строки, так и по идентификатору. Имелась возможность произвести групповой обмен (считывание всех входных данных на поле задачи), перемещать указатель в формуляре на любое число позиций, переходить на последовательный доступ, контролировать наличие запрашиваемого параметра и т.п.

Кроме формуляров с фиксированным числом строк («жесткий» формуляр) имелась возможность работы с «мягкими» формулярами, число строк в которых было заранее не определено и могло меняться в процессе работы проектных алгоритмов. Однако в этом случае обеспечивался только последовательный доступ к значениям параметров.

Каждый алгоритм проектирования снабжался паспортом, в котором перечислялись используемые входные и выходные формуляры. После загрузки соответствующей программы указанные формуляры извлекались в оперативную память из временной или постоянной библиотек, и только после этого управление передавалось вызванному модулю. После завершения его работы выходные формуляры возвращались на свое место во внешней памяти.

Комплекс проектировочных алгоритмов с описанным способом взаимодействия по данным вошел в состав программно-аппаратного обеспечения САПР «Чертеж», в 1977 г. был принят Междудементальной комиссией и введен в эксплуатацию.

5. РАСШИРЕНИЕ ИНТЕРАКТИВНЫХ ВОЗМОЖНОСТЕЙ ОПЕРАЦИОННОЙ СИСТЕМЫ ДОС ЕС

[5, 6, 27, 38, 42, 53]

Операционная система ДОС ЕС в течение длительного периода времени эксплуатировалась на младших моделях ЕС ЭВМ, которые были основным вычислительным средством большинства проектно-конструкторских организаций и научно-исследовательских институтов. Однако ДОС ЕС обладала серьезным недостатком - в ее составе отсутствовали не только какие-либо средства поддержания диалога с прикладной программой, не было даже такого необходимого инструмента для подготовки текстов программ, как редактора. В связи с этим в 1980-81 гг. была предпринята попытка создания полноэкранного редактора символьной информации на базе локальной дисплейной станции ЕС-7920. Предстояло создать программное обеспечение, не имевшее каких-либо аналогов или прототипов и удовлетворявшее ряду довольно жестких требований:

- естественность и наглядность процедур редактирования;
- высокая оперативность и хороший синтаксический контроль;
- устойчивость по отношению к сбоям ЭВМ и зависаниям операционной системы;
- многотерминальность при небольшом объеме используемой оперативной памяти;
- максимальное использование функциональной клавиатуры;

- возможность обслуживания дисковых библиотек с исходными текстами и различных внешних носителей информации (карты, магнитные ленты, АЦПУ).

Разработанный редактор был написан на АССЕМБЛЕРЕ и состоял из семи программных модулей, содержащих порядка 2500 операторов. Обмен с внешними устройствами был реализован на физическом уровне. Собственно редактор занимал в оперативной памяти порядка 14 Кб и динамически запрашивал для каждого активного пользователя дополнительный буфер емкостью в 6.7 Кб. Максимальное количество одновременно работавших пользователей не могло превышать 8, но ограничивалось оно не количеством дисплеев, а максимальным числом подзадач, которое поддерживалось в рамках ДОС ЕС.

Редактор предусматривал выполнение порядка 50 операций по вводу, редактированию и выводу текстовых документов различной природы.

Он обеспечивал работу с библиотеками исходных текстов ДОС ЕС - фиксацию имен дисков и библиотек, просмотр каталогов, извлечение документа или любого его фрагмента для редактирования или подключения к текущему файлу, удаление библиотечного модуля или запись нового и т.п.

В режиме редактирования пользователь мог обратиться к любому свободно-внешнему устройству, прочитать хранящуюся там информацию или записать фрагмент текущего набора, перемотать ленту на заданное число записей (строк) или файлов, распечатать документ или любую его часть.

Редактор включал ряд сервисных средств, связанных с набором исходных программ на том или ином языке программирования - вывод на экран измерительных линеек, табуляция курсора в заранее установленные позиции.

Наиболее употребительные операции, не требовавшие задания параметров, инициировались нажатием единственной функциональной кнопки - клавишными директивами (например, пропуск очередной строки при выборке из редактируемого файла, чтение очередной страницы, табуляция и т.п.). Другие процедуры набирались пользователем в командной строке на языке, близком к естественному - НАЙДИ такой-то фрагмент, ПЕРЕПИШИ столько-то строк и т.д. С целью сокращения времени набора часто используемые директивы допускали односимвольные сокращения - «К» вместо КАТАЛОГ, «З» вместо ЗАПИШИ и т.п.

Несмотря на более низкий рейтинг операционной системы ДОС ЕС по сравнению с ОС ЕС, редактор EDIT-7920 обеспечивал несравненно больший комфорт для пользователя и обладал гораздо более высоким быстродействием по сравнению с наиболее распространенной системой коллективного доступа PRIMUS. Время реакции редактора при максимальной загрузке редко превышало 1-2 с за исключением тех случаев, когда пользователь заказывал длитель-

ную операцию типа считывания большого массива с ленты или диска. Максимальный объем редактируемого документа составлял 10000 строк.

В случае аварийной перезагрузки операционной системы и повторного запуска редактора была предусмотрена возможность продолжить работу пользователей с минимальными потерями - можно было прочитать содержимое еще не погасшего экрана (точнее - автономной памяти дисплея) и начать редактирование с прерванной точки.

Тексты основных программных модулей редактора были опубликованы в монографии [6], что дало возможность читателям повторить аналогичную разработку самостоятельно. В период 1984-1987 гг. система редактирования в различных модификациях (ЕС-7927, ЕС-7906) была передана 7 организациям. Среди них НПО «Полимер» (г.Москва), НПО «Рубин» (г.Пенза), Горьковский телевизионный завод им. В.И. Ленина, Московский педагогический институт.

Одним из неожиданных внедрений редактора EDIT-7920 было его использование в подготовке исходных данных серии социологических исследований, проводимых Академией общественных наук при ЦК КПСС. Редактор позволил не только автоматизировать процесс ввода данных, но и существенно упростить процедуры внесения изменений, слияния массивов анкет разных обследований и выделения в них подмножеств, удовлетворяющих заданным условиям отбора. Технология обработки результатов анкетирования описана в препринте [5].

6. АДАПТАЦИЯ АЛГОРИТМИЧЕСКОГО ЯЗЫКА ФОРТРАН ЕС ДЛЯ СОЗДАНИЯ ДИАЛОГОВЫХ ПРОГРАММ

[4, 6, 26, 27, 28, 29, 30, 31, 33, 34, 35, 41, 50]

Пакет программ ДИАФОР-ЕС был разработан в 1980-82 гг. и предназначен для программирования диалоговых сценариев на ФОРТРАНЕ или АССЕМБЛЕРЕ ЕС ЭВМ. С его помощью создавались программы-посредники, обеспечивавшие диалог по запланированному сценарию между пользователем и его прикладной программой, выполняющейся в ЭВМ под управлением ДОС ЕС или ОС ЕС.

Отличительными особенностями этого пакета являются:

- максимальная гибкость процедур изменения значений варьируемых параметров и просмотра результатов вычислений, заключающаяся в предоставлении права выбора проектанту;
- возможность идентификации входных/выходных параметров в терминах профессионального языка без соблюдения жестких правил на фор-

мат данных и тип идентификаторов, налагаемых алгоритмическими языками;

- динамический выбор последовательности загружаемых проектных процедур;
- возможность активного вмешательства пользователя в ход вычислительного процесса.

Средства пакета позволяли не только программировать новые диалоговые комплексы, но и давали возможность осуществить с минимальными переделками перевод в диалоговый режим задач поискового характера (оптимизация параметров изделий, обработка экспериментальных данных и т.п.), ранее эксплуатировавшихся в пакетном режиме и требовавших многократных запусков на ЭВМ.

Диалоговые программы-посредники могли воспринимать сообщения пользователя на языке, близком к естественному, анализировать их по заранее фиксированным ключевым словам, изменять значения входных параметров прикладной программы, сообщать пользователю выборочные значения выходных параметров, менять схему вычислительного процесса, динамически загружая необходимые модули, реагировать на сигналы прерывания, поступающие с терминала по инициативе пользователя.

В зависимости от терминального оборудования и операционной системы, эксплуатируемых на ЭВМ, было разработано четыре версии пакета ДИАФОР:

1. ДИАФОР ДОС К - версия, работающая под управлением ДОС ЕС и использующая в качестве терминала операторскую консоль;
2. ДИАФОР ДОС Д - версия, работающая под управлением ДОС ЕС и использующая в качестве терминала дисплей ЕС-7927;
3. ДИАФОР ОС К - версия, работающая под управлением ОС ЕС и использующая в качестве терминала операторскую консоль;
4. ДИАФОР ОС Д - версия, работающая под управлением ДОС ЕС и использующая в качестве терминала дисплей ЕС-7927.

Пакет включал 38 подпрограмм, составленных на ФОРТРАНЕ и АССЕМБЛЕРЕ, и 6 макроопределений. Большую часть пакета (24 модуля) составляли программы и макроопределения, которые без каких-либо изменений входили в состав любой версии ПП ДИАФОР.

Общий объем пакета со всеми его модификациями - 1476 операторов на ассемблере и 301 оператор на ФОРТРАНЕ.

Версии пакета ДИАФОР-ДОС-К и ДИАФОР-ОС-К не предъявляли каких-либо требований к операционным системам и могли эксплуатироваться на любой серийной ЭВМ Единой Системы.

Версии ДИАФОР-ДОС-Д и ДИАФОР-ОС-Д работали под управлением операционных систем, в которых было предусмотрено функционирование средств

базисного телекоммуникационного доступа (БТМД). В составе периферийного оборудования таких комплексов должна была находиться дисплейная станция ЕС-7920 локального типа, к которой подключался хотя бы один дисплей ЕС-7927. Пакет ДИАФОР-ЕС мог использовать дисплейную станцию совместно с другими диалоговыми системами.

Модули всех версий пакета ДИАФОР-ЕС допускали обращение из программ, составленных на ФОРТРАНЕ или АССЕМБЛЕРЕ ЕС ЭВМ с учетом стандартных соглашений операционных систем о межпрограммных связях.

В зависимости от количества задействованных модулей и версии пакета ДИАФОР-ЕС программы, обслуживавшие сценарий диалога, занимали в оперативной памяти от 4 Кб до 10 Кб. Никаких других ресурсов ЭВМ, кроме терминального оборудования, модули пакета не использовали.

Первая апробация пакета ДИАФОР состоялась в 1980 г. на программных продуктах ПО «Гидромаш». Подсистема «Шимми» расчета устойчивости самолетных шасси, входившая в САПР предприятия, до этого эксплуатировалась в пакетном режиме. Для подбора оптимальных значений параметров изделия проектировщики были вынуждены производить большое количество поверочных расчетов, по результатам которых готовились очередные варианты. Пакет ДИАФОР позволил в кратчайшие сроки (порядка 3-4 недель) осуществить перевод расчетных программ в диалоговый режим, что существенно повысило оперативность принимаемых решений и сократило сроки подготовки производства новых изделий.

Среди других проектных организаций и промышленных предприятий, на которых внедрение пакета ДИАФОР производилось авторами, отметим Павловский механический завод им. С.И. Кадышева, ПМЗ «Восход» (г. Павлово), ПКТИ СХ «Оргснаб» (г. Горький), ИВЦ Сахалинского территориального геологического управления (г. Южно-Сахалинск). В некоторых автоматизированных системах, разработанных другими коллективами, в качестве диалоговой компоненты использовался пакет ДИАФОР. Автор работал в тесном контакте с создателями таких систем как САПР «Алмаз» и САПР «Авангард».

Тексты ключевых программ пакета были опубликованы в монографиях [4,6] и в полном объеме представлены в государственный фонд алгоритмов и программ [35]. Это позволило многим организациям повторить разработку без прямого обращения к авторам. Кроме того, в период 1981-1987 гг. документация по пакету была передана 13 НИИ, КБ и учебным заведениям.

7. ИНСТРУМЕНТАЛЬНЫЙ КОМПЛЕКС ДЛЯ ВОСПРОИЗВЕДЕНИЯ ИЗДАТЕЛЬСКИХ ОРИГИНАЛОВ КАРТОГРАФИЧЕСКИХ ДОКУМЕНТОВ

[39, 40, 41, 43, 44, 45, 47, 53, 54]

В период 1980-1990 гг. автор возглавлял работу по созданию ряда подсистем вывода картографических документов на ЕС и СМ ЭВМ. Работа эта являлась составной компонентой комплексных тем по автоматизированному вводу, распознаванию, хранению, автоматическому созданию и отображению картографических документов различной природы, выполнявшихся в НИИ ПМК под руководством д.т.н. Ю.Г. Васина.

По своему рангу эти работы носили, как правило, характер опытно-конструкторских и завершались созданием автоматизированных картографических систем - АСНИ соответствующего профиля. Внедрение указанных ОКР в практику работы подразделений Главного Управления Геодезии и Картографии, ЦКП ВМФ и других организаций заказчика позволило существенно сократить сроки подготовки картматериалов к изданию, повысить точность и качество издаваемых документов, автоматизировать процесс получения расчлененных по цветам оригиналов топографических и морских навигационных карт, планов и планшетов с лесоустроительной информацией.

В состав каждой АСНИ входил довольно крупный программный комплекс - подсистема ВЫВОД, предназначенный для получения высококачественных расчлененных по цветам оригиналов топографических и морских карт, удовлетворяющим требованиям редакционно-издательского отдела, а также контрольных копий различных картографических документов на любом этапе их создания.

В первом случае в качестве устройства вывода применяются прецизионные координатографы с векторным управлением типа КПА-1200 (разрешающая способность - 0.005 мм) или устройства растрового вывода графической информации на базе фототелеграфного аппарата ГАЗЕТА-2М (разрешающая способность - 0.04 мм). Вывод контрольных графических копий может быть осуществлен на любой графопостроитель или графический дисплей, поддерживаемый средствами ППП ГРАФОР.

Подсистема ВЫВОД функционирует на любой ЭВМ Единой Системы с объемом оперативной памяти не менее 1 мБ. Исходная информация, необходимая для работы подсистемы, включает заявку на обработку картографического документа, поступающую от пользователя в диалоговом или пакетном режиме, и цифровую модель карты, хранящуюся в оперативном архиве АСНИ. Диалоговый интерфейс с пользователем реализован в виде сценария на базе пакета ДИАФОР-ЕС.

В работе подсистемы ВЫВОД и ее настройке на предметную область (топографические, морские, тематические карты) существенную роль играет библиотека описаний картографических шрифтов и дискретных объектов. Для заполнения этой библиотеки объектами из соответствующей предметной области разработан специальный макроязык и создан оригинальный программный комплекс общего назначения.

Входной язык ориентирован на современные графопостроители, снабженные линейно-круговым интерполятором. Дополнительными особенностями языка являются учет толщины контурных линий и ее изменение при смене масштаба изображения, возможность заливки замкнутых областей, использование механизма аффинных преобразований, конструирование сложных описаний из уже имеющихся в библиотеке типовых графических элементов.

Подсистема ВЫВОД представляет собой сложный четырехуровневый программный комплекс. Верхний уровень подсистемы представлен управляющей программой, в задачи которой входят прием и анализ заказа на обработку, выборка из указанной базы данных очередного картографического объекта (КО), удовлетворяющего заданным условиям, анализ типа извлеченного объекта передача управления соответствующему интерпретатору.

Второй уровень подсистемы представлен интерпретаторами КО и их характеристик. Задача каждого интерпретатора заключается в декомпозиции сложных КО на составляющие компоненты и формировании всех графических характеристик, необходимых для воспроизведения элементов КО. Часть этой информации извлекается из соответствующих управляющих таблиц, в которых хранятся постоянные сведения о тех или иных характеристиках (тип шрифта и высота символов, код цветности, тип и толщина линии и т.п.). Другие сведения извлекаются из полей интегрального файла, представляющего очередной считанный КО. Для определения некоторых характеристик приходится выполнять довольно сложную вычислительную работу. Состав интерпретаторов КО и логика их работы при адаптации к новой предметной области меняются незначительно.

Третий уровень подсистемы ВЫВОД образуют комплексы исполнительных программ. Их задачей является формирование последовательности элементарных графических процедур, реализующих воспроизведение КО или их компонент виртуальным регистрирующим устройством.

В состав четвертого уровня подсистемы ВЫВОД включены процедуры формирования графической программы в терминах команд конкретного или условного графопостроителя.

В основу комплекса исполнительных программ положен пакет ГРАФОР, адаптированный для ЕС ЭВМ сотрудниками НИИ ПМК. Этот пакет, разработанный в ИПМ им. М.В. Келдыша АН СССР под руководством Ю.М. Баяков-

ского, долгое время был одним из наиболее популярных графических пакетов в нашей стране. В его состав входят порядка 400 программ, которые позволяют строить графики, гистограммы, карты изолиний, проекции поверхностей, применять методы сплайн-интерполяции и сглаживания, производить аффинные преобразования и экранирование, выполнять геометрические вычисления, осуществлять вывод графической информации на широкий спектр отечественных и зарубежных графопостроителей с векторным управлением.

Однако учет специфики картографических документов и требований, предъявляемых к издательским оригиналам, привел к существенной доработке пакета ГРАФОР. Основные направления его развития связаны со следующими моментами :

- разработка более эффективных методов кодирования метрической информации и переход от вещественной арифметики к целочисленным методам решения геометрических задач;
- разработка новых средств вывода символьной информации с учетом специфики картографических шрифтов;
- разработка методов кодирования и программ воспроизведения массового набора типовых графических элементов, представленных условными знаками топографических и морских карт;
- разработка программ воспроизведения разнообразных линейных знаков с учетом толщины линий, сложных законов чередования штрихпунктирных и других периодических заполнителей;
- разработка новых средств экранирования, обеспечивающих работу с большим количеством (порядка нескольких тысяч) экранируемых объектов;
- оптимизация времени выполнения графических программ, предназначенных для регистрирующих устройств с векторным управлением;
- разработка новых оконечных модулей для прецизионных координатографов типа КПА-1200 и растровых графопостроителей на базе фото-телеграфных аппаратов.

Общий объем программного обеспечения каждой модификации подсистемы ВЫВОД составляет порядка 25 тыс. операторов ФОРТРАНА и 3 тыс. операторов АССЕМБЛЕРА.

Личный вклад автора в создание серии подсистем ВЫВОД заключается в следующем:

- разработка архитектуры подсистем;
- разработка языка и технологии описания контуров дискретных условных знаков и символов картографических шрифтов;
- разработка сервисных программ воспроизведения условных дискретных и линейных знаков;

-
- разработка эффективных целочисленных алгоритмов решения геометрических задач;
 - разработка модуля вывода графической информации для прецизионного координатографа КПА-1200;
 - внедрение диалоговой компоненты на базе ППП ДИАФОР для формирования заявки на работу подсистемы ВЫВОД и редактирования графических характеристик КО в базе данных.

В 1990-1992 гг. автор руководил работами по переносу подсистемы вывода картматериалов в АСНИ «Морские карты» на персональные ЭВМ типа IBM PC. Им предложен новый подход к формированию контуров дискретных условных знаков и оригинальные программные средства отображения текстовых и графических документов на дисплеях и HP-совместимых графопостроителях, которые существенно расширяют возможности известного графического интерфейса VGI (Borland Graphic Interface).

8. ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

1. Разработана первая отечественная система диалогового программирования на базе языка БЕЙСИК. В рамках этой системы предложены и реализованы методы пошаговой компиляции и механизм быстрого переключения задач, обеспечившие многотерминальное обслуживание без тиражирования в памяти системных и сервисных программ. Разработанный подход заложил основы для создания многотерминальных систем автоматизированного проектирования.

2. Проведено исследование возможностей использования алгоритмического языка БЕЙСИК для создания диалоговых и графических компонент САПР и АСНИ. Предложены оригинальные средства включения в состав исходной программы фрагментов автокодного текста и процедур управления форматом вывода числовой информации, что позволило подключать к диалоговым программам ранее созданные библиотечные проектные процедуры и повысило качество оформления таблично-ориентированной документации.

3. Разработан многотерминальный редактор символьной информации в составе операционной системы ДОС ЕС, что позволило в интерактивном режиме создавать программное обеспечение САПР и АСНИ, просматривать и корректировать архивы с проектной документацией, осуществлять обмен информацией с внешними устройствами.

4. Предложена и реализована концепция естественного редактирования текстовых документов, разработан механизм предотвращения потери информации при сбоях и зависаниях операционной системы, что позволило повысить эффективность и результативность труда в системе «человек-ЭВМ».

5. На базе локальной дисплейной станции ЕС-7920 создан пакет ДИАФОР, обеспечивший возможность проектирования диалоговых сценариев в модулях САПР и АСНИ на языках ФОРТРАН и АССЕМБЛЕР ЕС ЭВМ. В составе пакета предложены средства изменения приоритета сторон, участвующих в диалоге, способы динамического управления логикой процесса выполнения проектных процедур.

6. Разработан комплекс программных средств для прецизионного вывода графических картматериалов, положенный в основу подсистемы документирования ряда АСНИ. Предложен графический макроязык для формирования библиотек с описаниями условных знаков и символов картографических шрифтов.

7. Разработанные инструментальные программные средства использованы для создания диалоговых и графических компонент в ряде отраслевых САПР и АСНИ, внедренных на промышленных предприятиях (ПМЗ «Восход», Павловский механический завод, Дмитровский завод фрезерных станков), в проектно-конструкторских и научно-исследовательских организациях (ОКБ ПО Гидромаш, ЦНИИ «Буревестник», НПО ПАРМА), в подразделениях Главного Управления Геодезии и Картографии, Центрального картпроизводства ВМФ, Поволжского лесостроительного предприятия.

Основные результаты диссертации опубликованы в следующих работах:

1. Задыхайло И.Б., Камынин С.С., Кетков Ю.Л., Корягин Д.А., Любимский Э.З., Шура-Бура М.Р. Математическое обеспечение. Операционная система БЭСМ-6. Монитор. Общее описание. Препринт ИПМ АН СССР, 1968. - 28 с.

2. Белослудцева М.М., Бочаров В.В., Ильин Э.Н., Кетков Ю.Л., Куракина И.М., Максимов В.С. BASIC - система пакетной обработки задач на ЭВМ типа БЭСМ-4, М-220, М-222. Учебное пособие под ред. Ю.Л.Кеткова. - Горький, изд. ГГУ, 1973. - 192 с.

3. Кетков Ю.Л. Программирование на БЭЙСИКЕ. - М.: Статистика, 1978. - 178 с.

4. Кетков Ю.Л., Максимов В.С., Рябов А.Н. Введение в системное программирование на языке АССЕМБЛЕРА ЕС ЭВМ. - М.: Наука, 1982. - 264 с.

5. Кетков Ю.Л., Кулагин А.С., Максимов В.С. Интегрированный вариант машинных программ для обработки эмпирических данных социологических исследований. - М.: изд. АОН при ЦК КПСС, 1983. - 82 с.

6. Зверев В.И., Кетков Ю.Л., Максимов В.С. Алфавитно-цифровые дисплеи ЕС-7920 в диалоговых системах. - М.: Наука, 1986. - 240 с.

7. Кетков Ю.Л. Диалог на языке Бейсик для мини- и микро-ЭВМ. - М.: Наука, 1988. - 368 с.

-
8. Кетков Ю.Л. GW-, Turbo- и Quick-BASIC на персональных ЭВМ типа IBM PC/XT и AT. - М.: Финансы и статистика, 1992. - 264 с.
 9. Кетков Ю.Л. Толковый словарь алгоритмического языка БЕЙСИК. - М. : Наука, 1992. - 324 с.
 10. Денисова С.Н., Кетков Ю.Л., Куркин Ю.С. Автоматизация расчетов уровней ожидаемой шумности в судовых помещениях на ЭВМ. «Судостроение», 8, 1970
 11. Задыхайло И.Б., Камынин С.С., Кетков Ю.Л., Корягин Д.А., Любимский Э.З., Штаркман В.С., Шура-Бура М.Р. Монитор ОС ИПМ. Труды ВКП-2, Новосибирск, 1970
 12. Елтаренко А.Н., Задыхайло И.Б., Зеленина Л.А., Зусман И.Х., Камынин С.С., Кетков Ю.Л., Любимский Э.З., Шелепов Н.А. Управление задачами и процессами в ОС ИПМ. Труды ВКП-2, Новосибирск, 1970
 13. Белослудцева М.М., Бочаров В.В., Ильин Э.Н., Кетков Ю.Л., Куракина И.М., Максимов В.С. Использование алгоритмического разговорного языка BASIC на ЭВМ типа БЭСМ-4. 2-я Всесоюзн. конф. по применению ЭВМ и мат. методов в планировании и управлении предпр., Горький, 1971
 14. Белослудцева М.М., Бочаров В.В., Ильин Э.Н., Кетков Ю.Л., Куракина И.М., Максимов В.С., Шилова Ю.В. Использование алгоритмического разговорного языка BASIC на ЭВМ типа БЭСМ-4, М-220, М-222. Симпозиум «Теория языков и методы построения систем программирования», Алушта, 1972
 15. Кетков Ю.Л. BASIC-система пакетной обработки задач на ЭВМ типа БЭСМ-4, М-220. 3-й конгресс математиков БНР, Варна, 1972
 16. Дутьшева Л.Я., Кетков Ю.Л., Максимов В.С., Манишин В.Т., Штейман Д.М. Математическое обеспечение системы разделения времени для ЭВМ М-222. Всесоюзн. конф. по автоматизации программирования ЭВМ типа М-20, Москва, 1973
 17. Кетков Ю.Л. Методические указания по изучению темы «Современные методы программирования». - М.: изд. ВЗИИТ, 1974. - 48 с.
 18. Кетков Ю.Л., Рябов А.Н. Расширение средств редактирования для управления выводом данных в BASIC-системе. Всесоюзн. конф. по математическому обеспечению ЭВМ типа М-20, Москва, 1975
 19. Дутьшева Л.Я., Кетков Ю.Л., Кузин С.Г. Обучение программированию в системе коллективного пользования «Студент». Сб. «Машинное обучение с помощью диалога», М., МДНТП, 1976
 20. Кетков Ю.Л., Рябов А.Н. Программирование на БЭЙСИКЕ: (Входной язык программирования системы БЭЙСИК - 222). Учебное пособие.- Горький, изд. ГГУ, 1977. - 73 с.

21. Будеков О.А., Гололобов А.В., Кетков Ю.Л. Эвристические алгоритмы одномерного размещения связанных элементов. Сб. «Выч. техника в машиностроении», Минск, 1977, вып. 2
22. Кетков Ю.Л., Максимов В.С. Организация и работа с данными в специализированной САПР. Всесоюзн. конф. «Матем. обеспечение моделирования сложных систем», Киев, 1977
23. Кетков Ю.Л. Разработка диалоговой системы на базе ЕС-1020 Респ. симп. «Теория и практика системного программирования», Винница, 1977
24. Кетков Ю.Л., Рябов А.Н. Эффективная система программирования BASIC - 222. «Программирование», 1, 1978
25. Кетков Ю.Л., Максимов В.С. Организация и работа с данными в специализированной САПР. Межвуз.сб. «Оптимизация и математическое обеспечение САПР», Горький, изд. ГГУ, 1978
26. Антипова Р.И., Кетков Ю.Л., Куркин Ю.С., Новин В.П. Автоматизация силового анализа конструкций на базе ЕС ЭВМ. 3-я Всесоюзн. школа «Матем. обеспечение АСУП», Горький, 1978
27. Кетков Ю.Л. Макрокоманды анализа символьной информации. ГосФАП, 1980, рег. номер П004370
28. Кетков Ю.Л. Диалог с прикладной программой под управлением ДОС ЕС. Всесоюзн.конф. «Методы мат. логики в проблемах ИИ и систематическое программирование», Паланга, 1980
29. Батищев Д.И., Кетков Ю.Л. Особенности архитектуры и языкового обеспечения систем генерации диалоговых ППП на ЕС ЭВМ. Всесоюзн. конф.»Автоматизация производства ППП», Таллин, 1980
30. Кетков Ю.Л. Программные средства для организации диалога с прикладной программой под управлением ДОС ЕС. Межвуз.сб. «Оптимизация и математическое обеспечение САПР», Горький, изд. ГГУ, 1980, с.126-140
31. Кетков Ю.Л., Куракина И.М. Математическое обеспечение графического диалога под управлением ДОС ЕС. Межвуз.сб. «Математическое обеспечение САПР», Горький, изд.ГГУ, 1981, с.31-39
32. Ильин Э.Н., Кетков Ю.Л., Максимов В.С. Опыт разработки системы программирования БЭЙСИК-ЕС. Всесоюзн.конф. «Методы трансляции», Новосибирск, 1981
33. Кетков Ю.Л., Куракина И.М. Архитектура диалоговой графической системы на ЕС ЭВМ. 4-я Всесоюзн. конф. «Интерактивная технология в САПР», Таллин, 1981
34. Ильин Э.Н., Кетков Ю.Л. Использование матричного переключателя для организации диалога с многофазовой программой под управлением ДОС ЕС. 4-я Всесоюзн.конф. «Интерактивная технология в САПР», Таллин, 1981

35. Кетков Ю.Л. Пакет программ ДИАФОР для организации диалога под управлением ДОС ЕС. ГосФАП, 1983, рег. номер П005892

36. Ильин Э.Н., Кетков Ю.Л., Максимов В.С., Максимова Ю.В. Основные характеристики диалоговой системы программирования БЭЙСИК-ЕС. 3-я Всесоюзн. конф. «ДИАЛОГ-83», Протвино, 1983

37. Ильин Э.Н., Кетков Ю.Л., Куркина Т.А., Максимов В.С., Тафориная Н.М. Расширение возможностей ФОРТРАНА ЕС ЭВМ для программирования диалоговых сценариев. 3-я Всесоюзн. конф. «ДИАЛОГ-83», Протвино, 1983

38. Кетков Ю.Л., Максимов В.С. Редактирование символьной информации в режиме диалога под управлением ДОС ЕС. 4-я Всесоюзн. конф. «Системное и теоретическое программирование», Кишинев, 1983

39. Гордион М.М., Кетков Ю.Л., Куракина И.М., Молоткова Т.Д., Тафориная Н.М. Расширение возможностей ППП ГРАФОР для воспроизведения полигональных линий и надписей. 1-я Всесоюзн. конф. «Методы и средства обработки сложной графич. информации», Горький, 1983

40. Антипова Р.И., Белова М.Л., Гордион М.М., Ильин Э.Н., Кетков Ю.Л., Куракина И.М., Молоткова Т.Д. Инструментальная подсистема для вывода картографической информации на ЕС ЭВМ. 1-я Всесоюзн. конф. «Методы и средства обработки сложной графич. информации», Горький, 1983

41. Ильин Э.Н., Кетков Ю.Л., Максимов В.С. Математическое обеспечение ЕС ЭВМ для организации обмена с периферийными СМ ЭВМ в локальной вычислительной сети. 1-я Всесоюзн. конф. «Методы и средства обработки сложной графич. информации», Горький, 1983

42. Кетков Ю.Л., Максимов В.С. Естественное редактирование текстовых файлов в диалоговом режиме под управлением ДОС ЕС. Межвуз. сб. «Математическое моделирование и программное обеспечение в САПР», Горький, изд. ГГУ, 1984, с.12-31

43. Гордион М.М., Кетков Ю.Л., Куракина И.М., Молоткова Т.Д., Тафориная Н.М. Программа воспроизведения ломаных линий и эквидистант на базе пакета ГРАФОР. Деп. сб. «Системные и прикладные программы», ВИНТИ, 4911-85 ДЕП

44. Кетков Ю.Л., Максимов В.С. Макроязык описания графических элементов. 2-я Всесоюзн. конф. «Методы и средства обработки сложной графич. информации», Горький, 1985

45. Антипова Р.И., Белова М.Л., Гордион М.М., Ильин Э.Н., Кетков Ю.Л., Куракина И.М., Молоткова Т.Д., Тафориная Н.М. Инструментальная подсистема для вывода картографической информации на ЕС ЭВМ. Межвуз. сб. «Автоматизация обработки сложной графической информации», Горький, изд. ГГУ, 1987, с.112-122

46. Кетков Ю.Л. Графические средства в алгоритмическом языке Бейсик. «Микропроцессорные средства и системы», 1, 1988, с.25-30
47. Васин Ю.Г., Башкиров О.А., Кетков Ю.Л., Кобрин Р.Ю., Кустов Е.А., Ясаков Ю.В. Автоматизированная система обработки информации морских карт (АСОИМК). 3-я Всесоюзн. конф. «Методы и средства обработки сложной графич. информации», Горький, 1988
48. Кетков Ю.Л. Три поколения систем программирования на базе языка Бейсик. Всесоюзн. конф. «Методы трансляции и конструирования программ», Новосибирск, 1988
49. Кетков Ю.Л. Размышления о БЕЙСИКЕ (три поколения популярного языка программирования). Межвуз.сб. «Математическое моделирование и оптимизация», Горький, изд. ГГУ, 1990, с.133-144
50. Пакет программ для организации диалога под управлением ДООС ЕС. Отчет НИИ ПМК по теме 03.7657, 1980
51. Редактор символьной информации на базе дисплейной станции ЕС-7920. Отчет НИИ ПМК по теме 03.0476, 1981
52. Перевод программ подсистемы «Шимми» САПР 1-ой очереди из пакетного режима в диалоговый. Отчет НИИ ПМК по х/д 03.7913 с ПО Гидромаш, 1980
53. АСНИ АКВАРЕЛЬ. Пакет программ ДИАФОР ЕС (программная документация). Отчет НИИ ПМК по теме 03.1392, 1985
54. Автоматизированная система обработки морских карт АСОИМК. Подсистема воспроизведения картографических документов ВЫВОД-АСОИМК (программная документация). Отчет НИИ ПМК по теме 03.8227, 1987

Она была первой

Проект первой ЦВМ был задуман в 1955-56 гг. коллективом сотрудников кафедры теории колебаний радиофизического факультета и активно поддержан ее новым и молодым еще тогда заведующим Н.А. Железцовым. Интерес к работам в этом направлении был определен его предшественником — академиком А.А. Андроновым, которому довелось участвовать в работе одной серьезной комиссии, проверявшей целесообразность расходования довольно больших средств в лаборатории управляющих машин. Выводы комиссия, сделала правильные, и впоследствии на базе лаборатории, возглавляемой членом-корреспондентом АН СССР И.С. Бруком, был создан Институт электронных управляющих машин (ИНЭУМ). Первый период более характерен работами теоретического направления, К ним относятся блестящая дипломная работа М. И. Фейгина, связанная с исследованием динамики поведения триггера (1952 г.), проект арифметического устройства ЭВМ последовательного действия (М.Я. Эйнгорин, 1954 г.), система команд и архитектура ЭВМ с двухуровневой памятью (А.М. Гильман, 1955г.).

Однако тот же М.И. Фейгин начал разработку первой в ГГУ цифровой лабораторной работы — специализированной машины со странным названием «7×7». Окончательное ее изготовление он довел в 1953 г. с бригадой студентов младше-то курса, в которую входили И.М. Клибанова, Е.Ф. Сабаев и А.В. Сергиевский. Машина «7×7» умела вычислять произведение трехразрядных двоичных чисел, и с ее помощью долгое время В.А. Дозоров наводил страх на студентов физмата и радиофака. Он подменял исправные элементы схемы на неисправные, а задачей обучающихся было локализовать ошибочный диод по таблице получающихся результатов.

Наряду с учебными макетами подобного рода и глубокими теоретическими изысканиями на кафедре началась и более кропотливая экспериментальная работа по созданию отдельных узлов и блоков цифровой техники, В 1954-55 гг. довольно много дипломных работ было посвящено решению этих практических задач. Исторически сложилось так, что выпускники кафедры, посвятившие себя ногой тематике, группировались вокруг А.С. Тарановича.

Основной объем работ по изготовлению блоков и монтажу машины ГИФТИ был выполнен в 1956-57 гг. За основу машины был принят проект А.М. Гильмана, однако, в процессе его реализации во многие узлы и блоки были внесены серьезные изменения. «Мозговой центр» в это время представляла группа инженеров-разработчиков, в которую, помимо А.С. Тарановича, входили А.М. Гончаров (выпускник 1955 г.), М.Д. Брейдо, Н.В. Жеглова, Г.Д. Зарницын и Р.Х. Садеков (все — выпускники 1956 г.). Кроме них в пуско-наладочных работах принимали участие группа техников и группа монтажников. К вы-

полнению отдельных работ, связанных с созданием машины ГИФТИ, были привлечены Г.Г. Денисов (блок управления пультовым «дисплеем») и В.И. Королев (блоки питания).

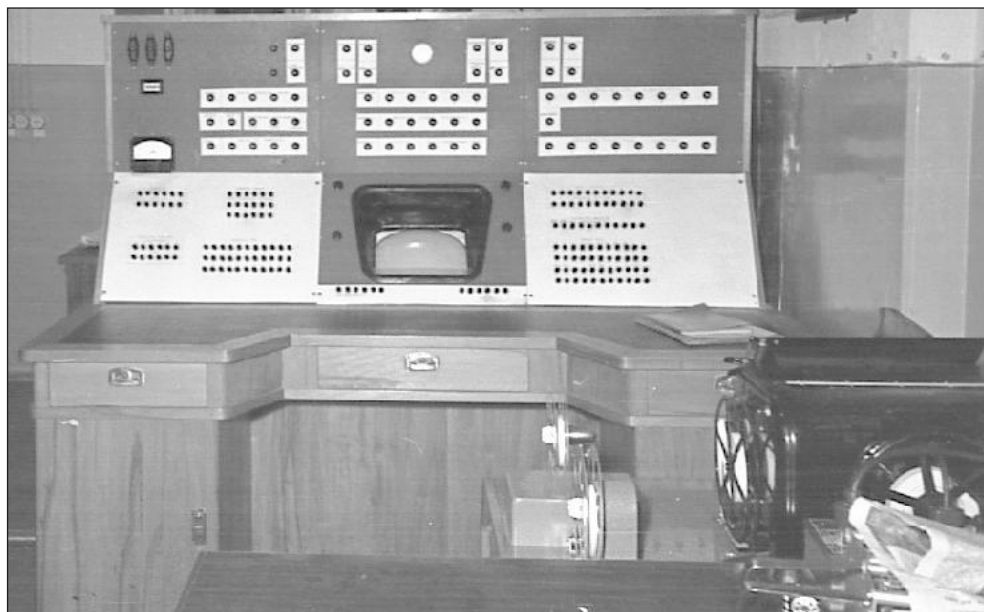
Для концентраций усилий по вводу машины ГИФТИ в строй и решения ряда задач организационного характера все эти коллективы в 1957 г. были подчинены доценту кафедры теории колебаний А.С. Алексееву. Принятые меры возымели действие, и к 40-й годовщине Октября машина «задышала».

Машина ГИФТИ обладала малым быстродействием – всего 100 операций в секунду, однако ее арифметический узел, несмотря на последовательную логику, мог выполнять до 6000 сложений в секунду.

Несмотря на сравнительно слабые эксплуатационные характеристики (малая скорость, небольшой объем оперативной памяти), в конструкцию машины ГИФТИ были заложены многие новинки, нашедшие широкое применение гораздо позже. Так, например, она имела сверхбыструю память, включавшую 32 рециркуляционных регистра. Система ее команд предусматривала три положения фиксированной запятой. В формате команд помимо трех адресов разной длины находились разряды, управляющие модификацией текущей команды и изменением содержимого регистра переадресации. Принятая в машине ГИФТИ идеология команд типа «регистр-память» сейчас широко используется в ЕС и СМ ЭВМ. На пульте машины находилась электроннолучевая трубка, предназначенная для просмотра содержимого ячеек оперативной памяти и используемая как своеобразный растровый дисплей. В арифметическом устройстве была реализована схема ускоренного умножения путем анализа двух соседних разрядов множителя.

Первый коллектив пользователей машины ГИФТИ состоял из трех программистов. Кроме автора этой заметки в него входили В.М. Корнилова и Ю.А. Первин. Мы помогали инженерам доводить конструкцию ЭВМ, составляли первые тесты, занимались разработкой программы нулевого цикла – подпрограмм ввода/вывода числовой информации, вычисления элементарных функций, реализации численных методов интегрирования, решения задач линейной алгебры, систем обыкновенных дифференциальных уравнений. К новому 1958 году на экране нашего дисплея появилась первая цифровая мультипликация: на елочки, контуры которых были образованы неподвижными битовыми комбинациями, опускались снежинки — перемещающиеся ярко светящиеся точки (двоичные «единицы»). Электронная докладная, сопровождающая этот графический шедевр и адресованная первому начальнику Вычислительного центра, гласила:

**АРТЕМИИ СЕРГЕЕВИЧ! ТРЕБУЕМ ПОВЫСИТЬ ЗАРПЛАТУ.
КЕТКОВ, ПЕРВИН.**



Первая машина ГИФТИ. Фото Е. КИПНИСА.

И начальник понял, что его подчиненные уже достигли уровня мировых стандартов.

До 1961 г. машине ГИФТИ была в Вычислительном центре единственной цифровой ЭВМ, на которой успешно решались многие научно-технические задачи и воспитывались первые кадры горьковских программистов.

Ю.Л. Кетков

К 30-летию ВЦ ГГУ, газета Горьковский университет (*примеч. ред.*)

Машина ГИФТИ

Проект первой в СССР вузовской цифровой вычислительной машины был задуман в 1955-56 гг. коллективом сотрудников кафедры теории колебаний радиофизического факультета ГГУ и активно поддержан ее новым заведующим Н.А. Железцовым, Интерес к работам в этом направлении был проявлен его предшественником – академиком А.А. Андроновым, которому довелось участвовать в работе одной серьезной комиссии, проверявшей целесообразность расходования довольно больших средств в лаборатории управляющих машин. Выводы комиссия сделала правильные, и впоследствии на базе лаборатории, возглавляемой чл.-корр. АН СССР И.С. Бруком, и разработанной там ЭВМ М-2 был создан Институт электронных управляющих машин (ИНЭУМ).

Первый период более характерен работами теоретического направления. К ним относятся блестящая дипломная работа М. Фейгина, связанная с исследованием динамики поведения триггера (1952 г.), проект арифметического устройства ЭВМ последовательного действия (М. Эйнгорин, 1954 г.), система команд и архитектура ЭВМ с двухуровневой памятью (А. Гильман, 1955 г.). Две последние работы были представлены на Всесоюзной конференции «Пути развития советского математического машиностроения и приборостроения», состоявшейся в 1956 г.

Воплощение двоичной логики в железо началось с цифровой лабораторной работы – специализированной машины со странным названием «7×7». Автором этого проекта был М. Фейгин, который довел свой замысел до ума в 1953 г. вместе с бригадой студентов младшего курса в составе И. Клибановой, Е. Сабаева и А. Сергиевского. Машина «7×7» умела вычислять произведение трехразрядных двоичных чисел, и с ее помощью ассистент В.А. Дозоров навел страх на студентов физмата и радиофака. Он подменял исправные элементы схемы на неисправные, а задачей обучающихся было локализовать ошибочный диод по таблице получающихся результатов.

Наряду с учебными макетами подобного рода и глубокими теоретическими изысканиями на кафедре началась и более кропотливая экспериментальная работа по созданию отдельных узлов и блоков цифровой техника. В 1954-55 гг. довольно много дипломных работ (С. Буторин, А. Гончаров, Б. Караулов, Б. Кожинская и др.) было посвящено решению этих практических задач. Исторически сложилось так, что выпускники кафедры, посвятившие себя новой тематике, группировались вокруг А.С. Тарановича (выпускника 1953 г.). В составе группы инженеров-разработчиков, включенной в штат ГИФТИ и активно поддерживаемой ее директором Я.Н. Николаевым, появились А.М. Гончаров (выпускник 1955 г.), М.Д. Брейдо, Н.В. Жеглова, Г.Д. Зарницын и Р.Х. Садеков (выпускники 1956-1957 гг.). Основной объем работ по изготовлению блоков

машины ГИФТИ выпал на группу, опекаемую З.С. Кечиевой. В составе великолепной семерки – монтажники А. Алексеев, А. Аралов, В. Блинничева, Л. Маркин, В. Монахов, А. Рожков и чертежница Д. Мануилова. В разработке силовых компонент (блоки питания, сетевые фильтры) принимал активное участие В.И. Королев. Дизайн пульта ЭВМ и разработка электронной схемы управления встроенным растровым дисплеем выполнены Г.Г. Денисовым. В 1957 г общее руководство работами по созданию, монтажу и вводу машины ГИФТИ в эксплуатацию было поручено к.ф.-м.н. А.С. Алексееву, который возглавил образованный к концу года Вычислительный центр ГИФТИ и руководил им, практически, до конца своей жизни.

В основу машины ГИФТИ был положен проект А.М. Гильмана, однако в процессе его реализации многие функциональные узлы подверглись серьезным изменениям. Машина ГИФТИ представляла собой универсальную ЦВМ последовательного действия с оперативной памятью из 2016 слов длиной по 32 бита. Специально для нее в ОКБМ был изготовлен магнитный барабан, вращавшийся со скоростью 6000 об./мин. На этом барабане помимо ячеек оперативной памяти были реализованы сверхбыстрые рециркуляционные регистры, позволившие довести скорость работы арифметического устройства до 6000 сложений в сек. Заметим, что ранее разработанные отечественные ЭВМ обладали быстродействием 100 оп/сек (Урал-1), 2000 оп/сек (Стрела-1) и 7000 оп/сек (БЭСМ). Причем в двух последних компьютерах была реализована более дорогостоящая параллельная арифметика. В арифметическом устройстве машины ГИФТИ была реализована схема ускоренного умножения и смоделирована оригинальная схема ускоренного деления двоичных чисел.

Общая производительность машины ГИФТИ сдерживалась медленной оперативной памятью. Однако система ее команд предусматривала довольно много операций типа регистр-регистр или память-регистр. Впоследствии такой подход стал основным в архитектуре машин третьего поколения – ЕС ЭВМ и СМ ЭВМ. Одной из оригинальных особенностей логики выполнения команд машины ГИФТИ была система автоматической модификации исполнительного адреса и управления приращением в индексном регистре. На пульте машины находилась электронно-лучевая трубка, предназначенная для просмотра содержимого регистров и ячеек оперативной памяти, используемая как своеобразный растровый дисплей. Заметим, что растровые дисплеи появились на ЭВМ третьего поколения почти через 20 лет. Ввод данных и программ производился с перфоленты со скоростью 300 знаков в сек тогда как электромеханический трансмиттер фототелеграфного аппарата выжимал всего 7 знаков в сек. Для вывода результатов вычислений использовался обычный рулонный телетайп РТА-50. По сравнению с ЭВМ, занимавшими тогда машинные залы в 200-300 кв.м. и потреблявшими по 200-300 квт электроэнергии, машина ГИФТИ пора-

жала своими скромными параметрами (комната в 25 кв.м, 560 ламп, потребляемая мощность – порядка 11 квт).

Таким образом, машина ГИФТИ оказалась пятой в стране универсальной цифровой вычислительной машиной вслед за ЭВМ БЭСМ (разработчик ИТМ и ВТ, руководитель акад. С.А.Лебедев, кстати, нижегородец по происхождению), Стрела-1 (разработчик Московский завод САМ, руководитель Базилевский), М-2 (разработчик лаборатория управляющих машин АН СССР, руководитель чл.-корр. И.С. Брук), Урал-1 (разработчик Пензенский завод САМ, руководитель Б.И. Рамеев). И она была первым компьютером, разработанным вузовскими учеными.

Первый коллектив пользователей машины ГИФТИ состоял из трех программистов. Кроме автора этой заметки в него входили В. Корнилова и Ю. Первин (выпускники физмата 1957 г.). Мы помогали инженерам доводить конструкцию ЭВМ, составляли первые тесты, занимались разработкой программ нулевого цикла – подпрограмм ввода/вывода числовой информации, вычисления элементарных функций, реализации численных методов интегрирования, решения задач линейной алгебры, систем обыкновенных дифференциальных уравнений. К новому 1958 г. на экране нашего дисплея появилась первая цифровая мультипликация: на фоне елочки, контуры которых были образованы неподвижными битовыми комбинациями, опускались снежинки – перемещающиеся ярко светящиеся точки (двоичные «единицы»). Электронная докладная, сопровождающая этот графический шедевр и адресованная начальнику Вычислительного центра, содержала требование повысить зарплату (зарплата программистов, зачисленных на должность и.о. старших научных сотрудников, всего в 3 раза превосходила их студенческую стипендию).

Первые инженерно-технические задачи, которые решались на машине ГИФТИ, были связаны с исследованиями систем обыкновенных дифференциальных уравнений высокого порядка. Главным поставщиком задач такого рода была лаборатория, возглавляемая Н.А. Железцовым, которая разрабатывала и исследовала схемы управления ядерными реакторами. Поначалу такие задачи решались на большой аналоговой вычислительной машине МН-8, многочисленные блоки которой соединялись в соответствии с математической моделью исследуемого объекта. Коммутация блоков осуществлялась вручную с помощью проводников на специальной панели, занимала много времени, требовала тщательной проверки и настройки параметров интегрирующих усилителей. Затраты на такого рода операции занимали несколько дней, и до тех пор, пока исследовалось поведение одной системы, коммутационную панель МН-8 нельзя было занимать для набора другой схемы. Погрешности аналоговых блоков и нестабильность их параметров не всегда гарантировали требуемую точность решения.

На машине ГИФТИ ввод программы решения аналогичной задачи занимал считанные секунды и после получения многометровых распечаток с таблицами исследуемых функций пользователь мог неспешно их анализировать, освобождая компьютер для решения других задач.

Следующую группу исследователей, вкусивших прелести численных методов решения дифференциальных уравнений, составили сотрудники Л.Н. Беллюстиной, занимавшиеся качественным исследованием динамических систем. Вскоре появились и первые внешние заказчики – проф. С.А. Жевакин (НИРФИ), проф. И.И. Трянин (ГИИВТ), молодые аспиранты проф. А.Г. Угодчикова (ГИСИ) и др. К 1960 г группа программистов в ВЦ ГИФТИ насчитывала в своих рядах уже более 20 чел.

До 1961 г. машина ГИФТИ была в Вычислительном центре единственной цифровой ЭВМ, на которой успешно решались многие научно-технические задачи и воспитывались первые кадры горьковских программистов.

Ю.Л. Кетков

Руководитель лаборатории программирования ВЦ ГИФТИ,
лауреат премии СМ СССР по кибернетике

01100 14 ↑	Прямое задание
01101 15 ↓	Обратное задание
00000 03 8-7.	Число, указанное в ИА вычитать по ИА.
00100 04 171	Сравнить числа a и b указанные в ИА и ИА по модулю. Если $ a > b $, то порядок следования команд нормальный; если сравнение не выполнено, то управление передается команде номер которой указан в ИА.
00101 05 ≡	$a \equiv b$
00110 06 >	$a > b$
00111 07 >	$a > b$
10000 10 814	Безусловная команда управления командой, номер которой указан в ИА.
00001 11 8	Арифметическое умножение
00010 12 V	Логическое сложение

01011 13 N	Изверсия
0110 16 17, 84	Формирование адреса к подпрограммам.
01111 17 811	
10011 23 ←	Сдвиг влево ($\times 2^{-1}$) К указателя в ИА
10100 24 ←	Сдвиг влево ($\times 2^1$)
10101 25 80, 7	Печать числа.
10111 27 86, 6	Ввод последовательности знаков
11100 34 86, 6	Ввод команды
11101 35 86, 7	Ввод десятичных (чисел).

Лекция №9

Такт машины — время, в течение которого выполняется одна команда. Главной целью.

Π_1 — выборка очередной команды. Номер ее счет в СК. Команда направляется в РТК.

Π_2 — переадресация: x содержимому РТК адресная часть содержимого указывается в команде переадресации. Переадресация команды отправляется в РТК.

Π_3 — выборка ИА

Π_4 — выборка ИА

Π_5 — арифметическая операция

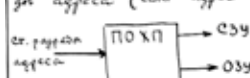
Π_6 — отслка результата

Π_7 — прибавление z_1 к СК (сохранение номера очередной команды).

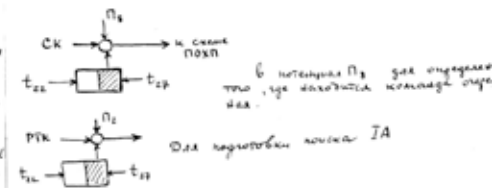
Π_8 — содержимое регистра Y_n применяется к соответствию с указанным в регистре Y_n . Если необходимо применяется содержимое СК.

Схема поиска в СЗУ.

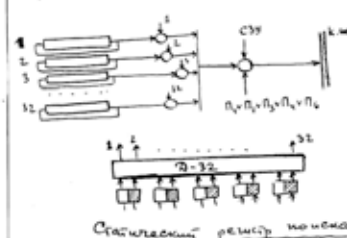
Принцип работы по поиску адреса осуществляется в СЗУ, является задачей нулевого или старшего разряда адреса (если адрес имеет 11-й разряд).



Работа схемы определяется характером поиска.



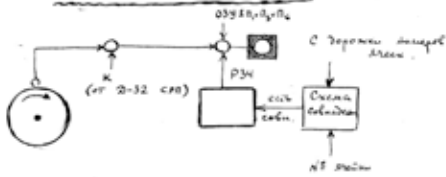
В Π_2 всегда есть сигнал Φ_{CSY} . Во время Π_2 в зависимости от того, что счет в ИА на ПОХП находится либо 0, либо ИА ($t_{21} - t_{22}$).



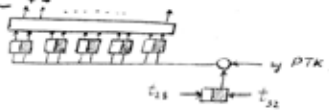
При поиске в СЗУ СРП расширяется номер регистра СЗУ, при поиске в ОЗУ или регистрами является номер программы.

Для этого используется сигнал с ПОХП, который только определяется в рег 3У. Соответственно и СРП задается те или иные разряды.

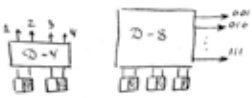
Схема поиска в УСУ.



Регистры кода описания, дешифратор кода описания.



Регистры γ_a и γ_b



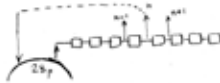
Выполнение γ_a



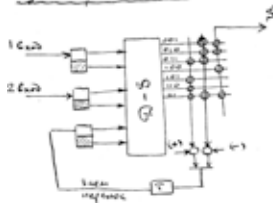
Функциональное устройство.



Функциональное устройство $P_2, P_1 - P_3$ представляет себе развешивающуюся регистр или МБ. За счет упорядочения или упорядочения этих регистров в них возможно выполнять код в ту или иную сторону.



Сумматор-вычитатель.



Модаль алгоритмический отращивание при различных операциях.

- ① Сложение
- ② Вычитание
- ③ Умножение
- ④ Деление

Схема Ф3Р, схема Ф3О

⑤ Операции сравнения

$$\begin{aligned} & \equiv a - b \stackrel{?}{=} 0 \quad [b] \neq 0 \\ |a| & \geq |a| - |b| \geq 0 \quad \text{если } a \text{ и } b \text{ знаки} \\ & \geq \text{sign } a + \text{sign}(a - |b|) \equiv 0 \pmod{2} \\ & > \left\{ \text{sign } a + \text{sign}(a - |b|) \equiv 0 \pmod{2} \right\} \& (|a| - |b| \neq 0) \end{aligned}$$

- ⑥ Логические операции.
- ⑦ Выбор команды
- ⑧ Выбор и выбор точек.

Школа программирования ИПМ им. акад. М.В. Келдыша

В конце 1967 г. Президиум АН СССР принял решение о разработке современного программного обеспечения БЭСМ-6, включающего многопользовательскую операционную систему, низкоуровневые (автокод) и высокоуровневые (алгоритмические языки) средства разработки и отладки программ. К исполнению проекта было разрешено привлекать квалифицированных системных программистов из разных организаций. В результате мне посчастливилось почти полтора года поработать в одном из самых мощных центров программирования в нашей стране. «Рекомендацией» к моему участию в этом проекте послужили годичная стажировка в ОПМ МИ АН СССР (1956-1957 гг.) во время производственной практики и выполнения дипломной работы, а также многолетние контакты с сотрудниками отдела программирования ИПМ (Э.З. Любимский был не только непосредственным консультантом по моей дипломной работе, но и оппонентом по защите кандидатской диссертации в 1965 г.).

Ниже приведен титульный лист препринта ИПМ, посвященный одному из центральных блоков операционной системы БЭСМ-6 – системе управления ресурсами, заданиями и процессами. К сожалению, из 7 исполнителей этого раздела на сегодня в живых остался только автор настоящего доклада. И я рассматриваю этот доклад как дань памяти моим коллегам – авторам ряда выдающихся пионерских работ в области системного и прикладного программирования.

АКАДЕМИЯ НАУК СССР
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
Отдел автоматизации программирования

ручная обработка
считывающего
ска предварито
точность ввода
вмест задания

МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ
ОПЕРАЦИОННАЯ СИСТЕМА " БЭСМ-6 "
МОНИТОР. ОБЩЕЕ ОПИСАНИЕ

Зав.отделом - М.Р.Шура-Бура

Исполнители: И.Б.Задыхайло
С.С.Камынин
Ю.Л.Кетков
Д.А.Корягин
Э.З. Любимский
В.С.Штаркман

Москва, март 1968 г.

1. М.Р. Шура-Бура (21.10.1918 – 14.12.2008)



Михаил Романович родился в селе Парафиевка Черниговской области в семье адвоката, которая из-за разрухи и голода в период гражданской войны переехала туда из Киева. В одном из интервью Шура-Бура так объясняет происхождение своей фамилии. Одна из слободок небольшого казацкого городка Борзна на Черниговщине носила прозвище Шура-Бура из-за буйного характера своих жителей (на местном наречии это название означало сильный ветер, бурю). Выходцев из этой слободки тоже награждали кличкой Шура-Бура. Возможно, что впоследствии за некоторыми из них и закрепилась такая фамилия. Вскоре после рождения Миши семья вновь вернулась в Киев, где прожила до 1933 г., а потом переехала в Москву. Здесь Миша закончил среднюю школу и в 1935 г. поступил на мехмат МГУ. После окончания МГУ он мечтал сразу поступить в аспирантуру, но встретил сопротивление со стороны партийного актива (не принимал участия в общественной работе, прогуливал физкультуру,

был замечен и в других прегрешениях). В студенческие годы о нем распевали куплет:

*На уроках физкультуры
Не бывало Шуры-Буры,
И за это Шура-Буре
Не бывать в аспирантуре.*

Поэтому в 1940 г. его направили на преподавательскую работу в Артиллерийскую академию. Однако его мечте суждено было сбыться: параллельно с работой в академии он в 1944 г. поступил в аспирантуру при НИИ математики МГУ. В 1947 г. аспирантура успешно завершилась кандидатской диссертацией по топологии (руководитель П.С. Александров), и его направили на работу в физико-технический факультет МГУ.

К началу 50-х гг. увлекся вычислительной математикой и в 1952 г. в соавторстве с Л.А. Люстерником, А.А. Абрамовым и В.И. Шестаковым опубликовал монографию «Решение математических задач на автоматических цифровых машинах. Программирование для быстродействующих электронных счетных машин». В 1954 г. это увлечение завершилось защитой докторской диссертации на тему «Вопросы решения математических задач с большим числом операций». Затем последовала краткосрочная рокировка: Физтех → Математический институт им. В.А. Стеклова → ИТМ и ВТ, где он нашел хорошее взаимопонимание с академиком М.А. Лаврентьевым, недавно утвержденным на посту директора. В ИТМ и ВТ Михаил Романович установил прочный контакт с инженерным отделом, который под руководством будущего академика С.А. Лебедева опекал вновь создаваемую ЭВМ БЭСМ и продолжал заботиться об ее предшественнице МЭСМ, эксплуатировавшейся в Киеве.

М.А. Лаврентьев предложил Михаилу Романовичу заняться проблемами расчета термоядерного взрыва (тогда эта задача была очень актуальной в связи с разработкой новых видов вооружения). А так как единственной более или менее устойчиво работающей ЭВМ располагало только вновь организованное Отделение Прикладной Математики при МИ АН СССР, то вычислительные эксперименты были перенесены на ЭВМ «Стрела-1». В ОПМ была выделена группа высококвалифицированных математиков под руководством К.А. Семендяева, а под крыло М.Р. Шуры-Буры попали молодые выпускники МГУ, работавшие в отделе программирования (тогда отдел возглавлял А.А. Ляпунов). Из-за неустойчивой работы ЭВМ расчеты по моделированию атомного взрыва приходилось разбивать на этапы, каждый из которых выполнялся не менее двух-трех раз с созданием контрольных точек, обеспечивавших продолжение эксперимента без возврата к ранее проведенным вычислениям. С трудом, но именно так объединенной группе удалось победить «Стрелу». За этот цикл работ в 1955 г. М.Р. Шура-Бура был удостоен Государственной премии

СССР (тогда она называлась Сталинской). Директор ОПМ академик М.В. Келдыш сумел рассмотреть в своем госте талант ученого и организатора, и после годичного притирания к группе программистов предложил Михаилу Романовичу формально возглавить отдел программирования, т.к. его фактическое руководство было уже признано. Новое место работы оставалось таковым уже до ухода Михаила Романовича из жизни.

Параллельно с руководством отдела в ОПМ Михаил Романович возвращается к преподавательской деятельности на мехмате МГУ. С 1955 г. он профессор кафедры вычислительной математики, а затем на протяжении почти четверти века (с 1970 г.) заведует кафедрой системного программирования. Под его руководством подготовлено 8 докторов наук и более 30 кандидатов. М.Р. Шура-Бура по праву был патриархом ведущей отечественной школы программирования. Подтверждением тому служат и многочисленные государственные награды: вторая Государственная премия (1978 г. – за подготовку и внедрение программно-аппаратных комплексов ЕС ЭВМ), орден Ленина (1990 г. – за создание бортового программного обеспечения беспилотного челнока «Буран»), ордена Трудового Красного Знамени (1956 г. – за разработку комплекса программ по расчетам параметров атомных бомб, 1983 г. – за создание программного обеспечения ЕС ЭВМ), орден «Знак Почета» (1961 г. – за разработку программ расчета траекторий искусственных спутников Земли), ряд медалей.

Попробую остановиться на некоторых научно-практических результатах работы отдела программирования. После ошеломительного успеха в области создания программного обеспечения задач атомной промышленности отдел Шуры-Буры наряду с созданием программ по заявкам других отделов ОПМ начал серию работ по совершенствованию труда программистов. Сначала появилась ПАПА – программа автоматического присвоения адресов (Т.А. Тросман, В.В. Мартынюк). Затем начали воплощаться в жизнь идеи операторного программирования, придуманные А.А. Ляпуновым. Разработка первой программирующей программы (ПП-1, 1954 г.) для ЭВМ «Стрела» велась С.С. Камыниным и Э.З. Любимским в порядке личной инициативы (она не входила в план работ отдела). Ее успех предопределил появление расширенного коллектива в лице И.Б. Задыхайло, В.С. Штаркмана, Э.С. Луховицкой, Т.А. Тросман и Т.П. Кузнецовой, которые помогли первопроходцам в создании ПП-2 (1955 г.). Эти программы заложили фундамент нового направления работ отдела в области автоматизации программирования. В 1956 г. я на себе испытал преимущества новых информационных технологий при выполнении дипломной работы.

Еще одной ступенькой, расширившей кругозор сотрудников отдела, стала работа по созданию библиотеки стандартных программ для ЭВМ М-20. Работу эту возглавил лично Михаил Романович, и здесь как нельзя кстати пришелся его опыт работы в области численного анализа. Интерпретирующая система

ИС-2 (позднее ИС-22) представляла маленький шедевр как по технологии выполнения библиотеки СП, так и по простоте ее использования, по минимуму накладных расходов. В этом же 1961 г. было принято решение о создании «Комиссия по эксплуатации вычислительных машин М-20», бессменным председателем которой стал Михаил Романович.

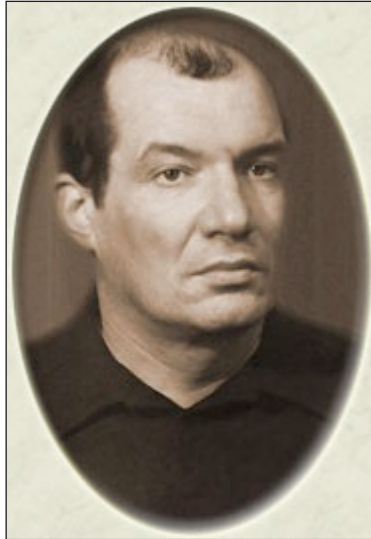
Очередной крутой подъем отделу программирования пришлось преодолеть на пути создания транслятора с полной версии языка Алгол-60 для ЭВМ типа М-20. Ограниченные ресурсы оперативной памяти способствовали появлению новых механизмов адресации. Появилась так называемая «математическая память», имевшая продолжение сначала на магнитных барабанах, а затем и на магнитных лентах. Математическая память была разбита на страницы, а в рамках компилятора действовала административная система, определявшая очередность подкачки и изгнания страниц из оперативного ЗУ. Этот механизм обусловил успешное прохождение транслятором ТА-2 международного теста «Man of You». **К созданию основной версии транслятора ТА-2 были привлечены лучшие силы отдела – главные конструкторы проекта Э.З. Любимский и С.С. Камынин, разработчики отдельных блоков И.Х. Зусман, Э.С. Луховицкая, В.В. Луцикович, В.В. Мартынюк, Г.М. Олейник-Овод, В.И. Собельман и Л.В. Ухов.** Кроме перечисленных штатных сотрудников в авторский коллектив удачно вписался молодой лейтенант Д.А. Корягин, прикомандированный в ОПМ из НИИ-4 Министерства обороны СССР.

1966-67 гг. характеризуются затишьем перед бурей – появлением одной из лучших ЭВМ того поколения БЭСМ-6. В одном из подразделений отдела под руководством В.С. Штаркмана ведутся работы по созданию автокода этой машины, удовлетворяющего всем современным требованиям к языку Ассемблера. По начальным буквам фамилий разработчиков эта версия получила название БЕМШ (Бочкова З.Ф., Езерова Г.Н., Морозова Л.Б., Штаркман В.С.). А в глубине другого подразделения нарастает революционный взрыв в индустрии создания трансляторов. С очередным проектом универсального алгоритмического машинно-ориентированного языка АЛМО выступил неугомонный тандем – С.С. Камынин и Э.З. Любимский. Идея совершенно очевидная: с появлением n машин разного типа и m алгоритмических языков разработчикам программного обеспечения понадобится написать $n*m$ трансляторов. Вместо этого предлагается написать m трансляторов с каждого алгоритмического языка на АЛМО, а затем еще n трансляторов с языка АЛМО в код каждой ЭВМ. Более того, каждый транслятор с языка высокого уровня сам пишется на АЛМО. За счет использования в АЛМО средств высокого и низкого уровней каждый переход требует сравнительно несложных алгоритмов преобразования. Успех этого проекта, обусловленный высоким профессиональным уровнем исполнителей и накопленным опытом создания первых средств автоматизации, пре-

взошел все ожидания. Несмотря на то, что аналогичный американский проект на базе языка UNCOL с треском провалился, отделу Михаила Романовича в ближайшие 2-3 года удалось создать серию двухступенчатых трансляторов, поставленных почти на все серийные ЭВМ отечественного производства. В частности именно таким образом у наших пользователей появился долгожданный ФОРТАН, прославившийся своими библиотеками и приложениями.

Конец 1967 г. и начало 1968 г. ознаменовался глобальным оснащением ЭВМ БЭСМ-6 современными средствами системного программирования – операционной системой, поддерживающей режим разделения времени, нормальными средами программирования как на низком (Ассемблер), так и на высоком уровне. По специальному постановлению Президиума АН СССР Отделению прикладной математики разрешили пригласить наиболее квалифицированных системных программистов из ведущих организаций страны. В разные периоды в составе этой сборной трудилось от 25 до 40-60 человек. Среди них были представители закрытых ядерных центров, конструкторских бюро, институтов Министерства обороны. В отделе Михаила Романовича полным ходом шли работы по созданию трансляторов, по разработке информационных систем и баз данных. Д.А. Корягин отвечал за систему логического управления файлами и периферией, И.Б. Задыхайло вместе с молодыми стажерами разрабатывал программы управления устройствами на физическом уровне, на мою долю пришелся монитор, который управлял ресурсами заданий, процессами и тактикой обслуживания различных очередей. Нити управления этим довольно сложным коллективом замыкались на наших корифеях – С.С. Камынине и Э.З. Любимском. Верхняя иерархия исполнителей, насчитывавшая 6-7 человек, приняла обет – не брить бороду или усы до сдачи системы. Все сотрудники ИПМ потешались, увидев в столовой нашу команду.

2. С.С. Камынин (10.01.1927 – 05.12 1986)



Сергей Сергеевич Камынин родился в Москве, в семье военных. В 1934 г. поступил в школу и к началу Великой Отечественной войны успел окончить 7 классов. После недолгой эвакуации вернулся в Москву, где с 1943 г. работал сначала учеником, а потом и слесарем по авиационному оружию. Школу окончил экстерном и в 1944 г. поступил на конструкторский факультет МВТУ им. Баумана. Учился там неровно, уделяя больше времени предметам, которые ему нравились. По этой тематике написал две работы, участвовавшие в городском студенческом конкурсе, отмеченные премией. Трижды отчислялся из института и в 1951 г. ушел окончательно, не закончив 2-й курс.

Академик Т.М. Энеев высоко оценил первую студенческую работу Камынина, посвященную реакции горения пороховой шашки для реактивного снаряда. По его мнению, уровень работы вполне соответствовал кандидатской диссертации. Энеев вместе со своим коллегой Д.Е. Охоцимским уговорили Камынина посещать семинары в отделе механики, которым в МИ АН СССР заведовал академик М.В. Келдыш. С 1951 г. его приняли в этот отдел на должность лаборанта (формальных оснований в приеме на инженерную ставку не было) и сразу же привлекли к выполнению научно-исследовательских работ. В первом же закрытом отчете 1951 г. по исследованию баллистических возможностей составных ракет (работы велись для КБ С.П. Королева) было указано 3 автора: М.В. Келдыш, Д.Е. Охоцимский и С.С. Камынин. Через пару лет появился следующий закрытый отчет, посвященный теоретическим исследованиям динамики полета составных крылатых ракет дальнего действия. С.С. Камынин

вошел в состав авторского коллектива вместе с М.В. Келдышем, В.А. Егоровым, Д.Е. Охочимским и Т.М. Энеевым. В открытой печати обе эти работы появились спустя 35 лет, уже после смерти самого молодого автора.

После создания ОПМ в 1953 г. и перевода туда отдела механики С.С. Камынин получил персональное приглашение нового директора, ценившего инженерный и исследовательский талант своего лаборанта. Однако вскоре после конфликта с Дмитрием Евгеньевичем С.С. Камынин перешел в отдел программирования, в котором проработал до конца своих дней. Правда, очень долгое время на должностях инженера и ст. инженера, для чего создавались многократные аттестационные комиссии и испрашивались рекомендации со стороны оstepененного научного персонала.

В разделе, посвященном М.Р. Шуре-Буре, уже упоминался ряд работ, выполнявшихся в период 1953-1970 гг. с участием С.С. Камынина. В 1956 г. он наряду с ведущими сотрудниками отдела программирования был награжден орденом Трудового Красного Знамени за создание программ расчета характеристик ядерного взрыва. Еще две государственные награды, совсем не соответствующие профессиональному уровню и научно-техническим достижениям выдающегося ученого, хранятся сегодня в семейной шкатулке – юбилейная медаль «За доблестный труд. В ознаменование 100-летия со дня рождения В.И. Ленина» (1970 г) и медаль «За трудовое отличие» за достигнутые успехи выполнения заданий XI пятилетки по развитию Советской науки и техники и внедрению результатов исследований в народное хозяйство (1986 г.).

В последующие годы Сергей Сергеевич увлекся проблемами искусственно-го интеллекта и робототехники. Вместе со своим учеником Ю.А. Бухштабом он предложил идею создания информационной системы «ВОПРОС-ОТВЕТ», которая могла бы осуществлять логический анализ базы данных для вывода новых фактов, не заданных в явном виде. В этой системе были предложены оригинальные алгоритмы устранения противоречий, возникающих в процессе вывода или изначально заложенных в базу данных. По быстрдействию разработанная система на порядок опережала лучшие зарубежные системы такого же класса.

Задача с роботами возникла в результате выполнения в ИПМ совместных работ с ИАЭ им. И.В. Курчатова. С.С. Камынину удалось решить задачу прецизионной автоматической сборки, когда точность позиционирования управляющих компонент робота была в десятки раз хуже требуемой точности сборки. Ближайшие сотрудники Сергея Сергеевича удивлялись, т.к. к такой задаче никто не приступался, заранее считая, что у нее решения не существует. В начале 80-х гг. по инженерным схемам и алгоритмам, разработанным под руководством С.С. Камынина, был создан адаптивный многоцелевой промышленный робот РК-1. Его высокие эксплуатационные качества отмечены в книге

бывшего министра атомной промышленности О.Е. Адамова «Не благодаря, а вопреки».

Разительный контраст между профессионализмом С.С. Камынина и размером штатной ставки, которую он занимал в течение многих лет, привел к попытке представить Сергея Сергеевича к званию почетного доктора физико-математических наук «honoris Causa», не требовавшего свидетельства о высшем образовании. Очень активную поддержку инициативе ученого совета ИПМ оказали академики В.М. Глушков, А.А. Дородницын, А.А. Самарский, члены-корреспонденты А.П. Ершов, С.С. Лавров. 14 июня 1978 г. на ученом совете ИПМ было принято решение о соответствующем представлении в Президиум АН СССР. Однако 24 июня скоропостижно скончался М.В. Келдыш – директор ИПМ и одновременно Президент Академии наук. Без его личной поддержки решение совета воплотить в жизнь не удалось. Правда, на ближайшей перепечатки С.С. Камынин был утвержден в должности старшего научного сотрудника без ученой степени.

Яркий генератор новых научных направлений, большинство из которых доводили до воплощения его ученики, отличался научной скромностью и не спешил застолбить свои идеи в публикациях. Большинство из его немногочисленного списка работ появилось в препринтах ИПМ. Две основные монографии с результатами последних исследований были подготовлены сотрудниками ряда отделов ИПМ уже после смерти С.С. Камынина. Первая из них, посвященная работам по машинному зрению, вышла в 1988 г. под редакцией Д.Е. Охочимского. Вторая появилась в 1992 г. и включала основные законченные результаты и черновые заметки по принципам проектирования и разработки многофункциональных программируемых роботов. Она была подготовлена по инициативе одного из его лучших друзей – Д.А. Корягина.

3. Э.З. Любимский (25.11.1931 - 13.02.2008)



Эдуард Зиновьевич родился в Москве в семье военного инженера. Поступил в школу, затем война, эвакуация, суворовское училище. Вернулся в Москву, в 1949 г. закончил школу с золотой медалью, затем мехмат МГУ по кафедре вычислительной математики. Прослушал курс по основам программирования, который читал Алексей Андреевич Ляпунов. Заинтересовался языком операторных схем и даже попытался предложить проект программирующей программы, построенной на базе этого языка.

Любимского всегда отличали логическая последовательность в действиях, умение до конца разобраться в любой ситуации, просто и доходчиво излагать свои мысли, желание всегда доводить до конца начатое дело. Он был доброжелателен, редко повышал голос, умел правильно выстроить взаимоотношения со своим собеседником. Несмотря на то, что в 1956 г., когда я проходил в ОПМ

преддипломную и дипломную практику, моим формальным руководителем был А.А. Ляпунов, настоящим шефом оказался Саша Любимский. Так его называли друзья и коллеги по отделу программирования. И только за пределами этого круга он был известен как Эдуард Зиновьевич. Очень проникновенные воспоминания о Саше оставил Д.А. Корягин, посвятив их памяти Любимского. Встретившись в последний раз с тяжело больным другом, Д.А. Корягин сказал: «Сашка, ты всегда и навсегда со мной».

Э.З. Любимский был первым программистом, защитившим в СССР кандидатскую диссертацию по своей прямой специальности. Это событие произошло в 1957 г., работа называлась «Об автоматическом программировании и методе программирующих программ». Обстановка в научном сообществе тогда была довольно сложная, еще сказывались отзвуки жестокого преследования кибернетики как лженауки. «Чистые» математики (теоретики) тоже пока не почитали программирование за достойную науку – в работах подобного рода кроме кропотливого труда и первых находок, превращавших программирование из ремесла в искусство, не существовало теорем, доказательств, условий необходимости и достаточности. И только благодаря огромной поддержке со стороны руководства ОПМ (академик М.В. Келдыш, профессора А.А. Ляпунов, М.Р. Шура-Бура, руководители других научных отделов ОПМ), а также отзвукам недавно прошедших наградений за исследование характеристик ядерного взрыва чаша весов склонилась в пользу диссертанта. Хотя до утверждения решения Совета в ВАКе прошло почти 3 года. Может быть, меньшие сложности произошли и в период подготовки к защите докторской диссертации Э.З. Любимского на тему «Возможности и принципы построения операционной системы для БЭСМ-6 (ОС ИПМ)». Представлена она была к защите в 1971 г., но защиту пришлось провести только через два года и не в Москве, а в институте Кибернетики АН УССР (г. Киев).

С 1954 г. Э.З. Любимский совмещает научно-исследовательскую работу в ОПМ с преподавательской деятельностью в МГУ. Для студентов 1 и 2 курса мехмата читает лекции по программированию, сначала в должности ассистента, затем (с 1970 г.) – доцента. В 1978 г. становится профессором кафедры системного программирования факультета ВМК МГУ. Среди его учеников 45 кандидатов и 5 докторов наук. В 2001 г. ему присвоили звание заслуженного деятеля науки Российской Федерации.

Карьерный рост в ОПМ протекает сравнительно спокойно по отлаженной лестнице: сначала (до 1962 г.) младший научный сотрудник, затем кратковременное пребывание в должности старшего научного сотрудника и до 1988 г. – заведование сектором. В 1968 г. во время разработки ОС ИПМ на Э.З. Любимского временно легли обязанности зав. отделом программирования (в это

время М.Р. Шура-Бура тяжело заболел, – инфаркт, и долго возвращался к нормальному состоянию). С 1988 г. Э.З. Любимский заведует отделом ИПМ.

Очень большую роль в жизни Э.З. Любимского сыграла его дружба с С.С. Камыниным. В близком кругу единомышленников этому тандему был присвоен идентификатор ЛЮКС. Любимскому всегда удавалось обуздать и успокоить своего друга, который обладал не в меру вспыльчивым характером. Завершающая формулировка гениальных идей после бурного обсуждения проходила более спокойную редакторскую правку и становилась абсолютно прозрачной для коллектива исполнителей.

Любимский был очень жизнерадостным человеком, принимал активное участие в подготовке и проведении праздничных и юбилейных капустников, любил музыку (особенно И.С. Баха), собирал альбомы известных и не очень известных художников, вместе с женой Ириной коллекционировал изделия хохломской росписи, сочинял стихи.

Последний период его работы был связан с созданием инструментально-базового комплекса (ИБК) для системы редакционно-издательской подготовки и выпуска газет. Все это начиналось с нуля на базе издательства газеты «Правда». Э.З. Любимскому удалось сплотить вокруг себя преподавателей кафедры системного программирования факультета ВМК, аспирантов и энтузиастов по прежним проектам. Первоначальный программный комплекс для издательства перерос в гораздо более универсальный проект для разработки документно-ориентированных информационных систем и информационной поддержки крупных организаций.

Среди наиболее важных публикаций отметим учебное пособие для вузов по программированию (1980 г.), написанное в соавторстве с В.В. Мартынюком и Н.П. Трифоновым, и серию книг издательства «Мир» по математическому обеспечению ЭВМ, регулярно выходившую в свет под редакцией Э.З. Любимского. Большинство из них было переведено сотрудниками отдела программирования ИПМ.

4. В.С. Штаркман (16.10.1931– 21.02. 2005)



Всеволод Серафимович Штаркман родился в Москве. Он один из первой когорты выпускников МГУ 1954 г., который посвятил свою жизнь становлению и развитию программирования в нашей стране. Поступив в ОПМ по распределению, он проработал в этом институте, начиная с должности младшего научного сотрудника до заведующего отделом системного программирования.

По словам его младшего брата, Сева был удивительным ребенком, говорить научился раньше, чем ходить. Необыкновенно подвижный и любознательный, увлекался разнообразной техникой (радиолюбительство, авиамоделизм). Свободное от школы время проводил в дворовом гараже, помогая механикам ремонтировать автомобили. В школе учился только на отлично. Любил посещать лекции в Политехническом музее, и после окончания школы был принят

на физтех МГУ. Но после 1-го курса ему пришлось сменить факультет (проклятая 5-я графа в личном деле) и доучивался на мехмате в одной группе с Любимским и Задыхайло. Его увлечение техникой сказалось и на первых этапах профессиональной работы. Он досконально изучил систему команд ЭВМ «Стрела-1», много общался с инженерным персоналом, профессионально освоил логические и электронные схемы разных узлов и блоков. Заново написал систему тестовых программ, которые проверяли машину намного тщательнее, чем штатное тестовое обеспечение. Первое пособие по системе команд «Стрелы» Сева написал будучи на 5-м курсе, когда проходил практику в ОПМ. В технической документации такую «мелочь» не предусмотрели, и все будущие программисты начинали свое образование с этой библии.

Погружение в компьютер на физическом уровне привело к тому, что в ряде проектов последующих ЭВМ Севу приглашали в качестве эксперта не только по формированию набора команд, но и по принятию некоторых конструкторских решений. Он участвовал в проектировании ЭВМ М-20, Восток, Весна и ряда оборонных комплексов. Для некоторых из них занимался разработкой системного программного обеспечения, низкоуровневых трансляторов машинных команд (ассемблеры) и более продвинутых алгоритмических языков. Именно в его секторе начиналось освоение программного обеспечения машинной графики. Среди популярных разработок его учеников – пакет ГРАФОР (Ю.М. Баяковский, Т.Н. Михайлова), автокод БМ-4/220, транслятор с Фортрана Ф-20.

Студенческие годы и период работы в ОПМ характерны увлечением различными видами спорта – горные и водные лыжи, большой и малый теннис. В МГУ В.С. Штаркман входил в состав сборной факультета по волейболу. Будучи одним из первых автолюбителей в ОПМ, вывозил сотрудников своего сектора на водные акватории и приучал их держаться на буксире за катером.

Огромный вклад в развитие отечественного программирования В.С. Штаркман внес переводами монографий лучших зарубежных ученых. Серия книг по структуре ИВМ/360/370 и системе их команд, по алгоритмическим языкам С, С++, по ассемблерам и операционным системам. Ему вместе с Михаилом Романовичем пришлось немало потрудиться по адаптации программного обеспечения ЕС ЭВМ в нашей стране.

5. Д.А. Корягин (05.06.1935 – 23.6.2009)



Дмитрий Александрович родился в г. Уссурийске Приморского края. После суворовского училища поступил в Военно-воздушную инженерную академию им. Жуковского, которую окончил в 1959 г. Молодой лейтенант получил назначение в военное НИИ и сразу был прикомандирован в ОПМ для освоения библиотеки стандартных программ М-20. Был тепло принят коллективом отдела программирования, поближе познакомился со старейшинами, успевшими заслужить первые правительственные награды, – с Э.З. Любимским, впоследствии ставшим его научным руководителем кандидатской диссертации, с С.С. Камыниным, с И.Б. Задыхайло, с В.С. Штаркманом, который принимал активное участие в разработке системы команд М-20 и набора тестовых программ. И, наконец, застал начальную стадию работ по созданию библиотеки стандартных программ, выполнявшуюся под руководством и при непосредственном участии М.Р. Шуры-Буры.

После завершения первой командировки Д.А. Корягин несколько раз приезжал в ОПМ для консультаций по вопросам программирования. Вскоре по согласованию с руководством НИИ-4 был вновь прикомандирован в отдел программирования не только для повышения квалификации, но и для участия в большом проекте по созданию транслятора ТА-2 с алгоритмического языка АЛГОЛ в код машин типа М-20. Накопленный опыт работы в передовом про-

граммистском коллективе вылился в кандидатскую диссертацию «Транслятор с языка Алгол для ЭВМ М-50», защищенную в 1965 г.

Следующая более длительная командировка в ИПМ была связана с разработкой системного программного обеспечения БЭСМ-6. Д.А. Корягин вступил в упоминавшуюся ранее команду уже в чине майора с ученой степенью кандидата технических наук. Здесь, собственно, и состоялось наше первое знакомство, затянувшееся на долгие годы. У нас был общий кабинет в ИПМ, в котором довольно много времени проводил И.Б. Задыхайло. Почти каждый день нас посещали корифеи проекта – С.С. Камынин и Э.З. Любимский, обсуждая текущие проблемы и корректируя, по мере надобности, функциональность тех или иных модулей системы. Во время одного из перекуров мы выяснили, что во время эвакуации вместе с Д.А. Корягиным попали в один из поселков Чкаловской области со странным названием Кувандык. А т.к. мы с ним одногодки (Дима старше меня всего на один месяц), то вполне вероятно, что учились либо в одном, либо в параллельных классах единственной на весь поселок железнодорожной школы.

Дослужившись до чина полковника, Д.А. Корягин вышел в отставку и перешел на работу в ИПМ. Вскоре у него здесь появился сначала сектор, а потом и собственный отдел. В 1984 г. работа по созданию пакетов программ для решения задач математической физики (САФРА) завершилась защитой докторской диссертации. Вскоре он занялся проблемой автоматизации проектирования и изготовления машиностроительных изделий. Одним из первых проектов такого назначения была система КАПРИ, создававшаяся по заказу ИАЭ им. Курчатова. Успешное внедрение этой системы обусловило появление нового проекта – ИРБИС (Интегрированная Разработка Больших Инженерных Систем), выполнявшаяся совместно с ведущим институтом Минатома НИКИЭТ. Она в значительной мере упорядочила организацию работ по проектированию и расчетам трубопроводов атомного реактора. Директор НИКИЭТ О.Е. Адамов с большой теплотой отзывается о тех работах и личных качествах Д.А. Корягина. В 1986 г. Дмитрий Александрович был удостоен премии Совета Министров СССР.

В начале 90-х Д.А. Корягин выполнил цикл работ по созданию информационно-вычислительной сети ИПМ, на двух удаленных площадках которого функционирует более 200 персональных компьютеров и серверов. Эта работа нашла свое продолжение в разработке Grid-технологий, предусматривающих совместное использование сетевых ресурсов и реализацию распределенных вычислений. Последней строкой в его служебной карьере была должность заместителя директора ИПМ.

Научно-исследовательскую работу в ИПМ Д.А. Корягин успешно сочетал с преподавательской деятельностью на кафедре системного программирования

ВМК МГУ. В 1987 г. он был избран на должность профессора. Читал лекционный курс по пакетам прикладных программ. Под его руководством защищено 10 кандидатских диссертаций.

6. И.Б. Задыхайло (11.02.1931 – 02.08.1998)



Игорь Борисович родился в г. Белгород, там же в 1949 г. окончил школу и поступил на механико-математический факультет МГУ. В 1953 г., будучи студентом 5-го курса, был принят на работу в ОПМ. Сначала в качестве лаборанта участвовал в создании программ для ЭВМ Стрела-1 для расчета параметров ядерных взрывов. После окончания МГУ в 1954 г. был переведен на должность младшего научного сотрудника и уже в 1956 г. получил свою первую правительственную награду – орден Трудового Красного Знамени в связи с успешным завершением проекта по созданию водородной бомбы. В этом же

году мы, студенты Горьковского университета, – предстоящий первый выпуск математиков-вычислителей, – познакомились с сотрудниками отдела программирования. Одному из нас (Ю.А. Первину) посчастливилось попасть под опеку И.Б. Задыхайло, под руководством которого была написана первая игровая программа для ЭВМ Стрела-1. Эта программа моделировала игру в «морского козла» (разновидность партий в домино). Защита дипломной работы состоялась по всем правилам при огромном стечении болельщиков. В паре с компьютером играл автор программы, в качестве противников выступала сборная ОПМ. В первой же партии электронный партнер сделал крупную рыбу (более 60 очков), благодаря чему дипломная работа удостоилась оценки «отлично».

После ядерных расчетов И.Б. Задыхайло привлекли к новой, не менее амбициозной программе, связанной с расчетами параметров космических траекторий, проводившихся в отделе будущего академика Д.Е. Охоцимского. Для преодоления неустойчивой работы ЭВМ и достаточно скудных аппаратных ресурсов оперативной памяти в этих программах приходилось принимать меры по резервному копированию отдельных этапов алгоритма и оптимизации вычислений за счет использования специфики групповых операций ЭВМ «Стрела». Успешное выполнение очередного этапа отмечено медалями «За участие в первом полете человека в космос» и «За участие в первом выходе человека в космос», которые добавились к ранее полученному ордену.

Первое увлечение системным программированием началось с участия И.Б. Задыхайло в проекте по созданию второй версии программирующей программы ПП-2 (1955 г.). Идея автоматической трансляции программы по операторной схеме А.А. Ляпунова и опыт программирования сеточных методов в задачах математической физики послужили толчком к написанию статьи, в общем-то теоретического характера, по организации циклического процесса счета по параметрической записи специального вида (Журнал вычислительной математики и математической физики, 1963). В каком-то смысле эта работа была предшественником параллельных вычислений, опередившая на десяток лет появление аналогичных исследований за рубежом. Кандидатская диссертация, развивавшая эту тему, была защищена в 1964 г. Еще раз к многопроцессорным системам и параллельным вычислениям И.Б. Задыхайло вернется лет через 10, когда в ИПМ начнет появляться соответствующее компьютерное железо – ПС-3000, матричные процессоры, ЕС-1191. В этот период можно выделить три новых направления исследований, в которых ведущая роль принадлежит Игорю Борисовичу.

Во-первых, это развитие непроцедурных средств параллельного программирования, завершившееся созданием языка Норма и транслятора с этого языка для обеспечения задач математической физики (препринты ИПМ, 1985, 1990). Во-вторых, это развитие средств параллельного программирования в языке

Фортран (ФОРА-ЕС для ПС-3000, Фортран для СуперЭВМ). Наконец, третье направление связано с вопросами проектирования узлов и блоков многопроцессорных ЭВМ, которое развивалось в партнерстве с инженерным отделением ИПМ (А.Н. Мямлин). Результаты этого сотрудничества нашли свое отражение в новых инженерных решениях архитектуры третьего ряда ЕС ЭВМ. В 1983 г. за выполнение работ по созданию программного обеспечения ЕС ЭВМ Игорь Борисович был награжден медалью «За трудовое отличие».

По крайней мере для меня очень памятным было участие И.Б. Задыхайло в проекте по созданию операционной системы ОС ИПМ (1967-1970 гг.). Почти полтора года в этот период Д.А. Корягин, я и И.Б. Задыхайло работали за соседними столами. Игорь Борисович опекал плеяду будущих коллег – дипломников И.А. Бахарева, В.А. Крюкова, Л.А. Позднякова, С.А. Усова. Под его руководством разрабатывались программы обслуживания периферийных устройств, основной модуль супервайзера (надсмотрщика за внешним хозяйством БЭСМ-6). Среди 37 препринтов ИПМ, выпущенных по программной документации БЭСМ-6, в 9-ти И.Б. Задыхайло является соавтором.

С 1972 г. И.Б. Задыхайло по совместительству работал доцентом на кафедре системного программирования ВМК МГУ. Под его руководством защищено более 10 кандидатских диссертаций.

В качестве заключительного аккорда я приведу фотографию 1969 г., на которой представлена часть команды ОС ИПМ, давшая обет не сбривать бороду и/или усы до момента запуска в эксплуатацию операционной системы:



Слева направо – Э.З. Любимский, Д.А. Корягин, Ю.Л. Кетков, И.Б. Задыхайло и Л.В. Ухов у нового входа в здание ИПМ АН СССР.

Из материалов SORUCOM-2011. Вторая Международная конференция «Развитие вычислительной техники и ее программного обеспечения в России и странах бывшего СССР». 12 – 16 сентября 2011, Великий Новгород (*примеч. ред.*)

Как преодолеть барьер Windows?

Каждая операционная среда предъявляет определенные требования к оформлению прикладных программ. Пользователи ЭВМ третьего поколения до сих пор с ужасом вспоминают мытарства перехода от среды малых операционных систем типа ДОС-ЕС к большим операционным системам ОС-ЕС. На одной из всесоюзных конференций по программированию злые языки расшифровали аббревиатуру ОС как «огромная стена», стоящая перед пользователем. Язык управления заданиями JCL (Job Control Language) для многих становился непреодолимым препятствием, гораздо более сложным, чем освоение нового алгоритмического языка. Необходимость запоминания многочисленных параметров и их комбинаций в командах управления, описания структур данных за пределами программы, назначения и переназначения устройств, анализ многочисленных (свыше 5000) сообщений на иностранном языке – вот далеко не полный перечень потока дополнительной информации, обрушившейся на головы непрофессиональных пользователей.

С аналогичными проблемами, но уже на уровне внутренней организации программ, мы сталкиваемся сегодня при переходе от сравнительно простой операционной системы MS-DOS к более современным операционным средам типа Windows 3.x/95/98/NT. Учебные курсы на базе традиционных языков программирования типа Фортран, Бейсик, Паскаль или Си делали больший акцент на выработке алгоритма и соответствующей ему машинной программе в стиле процедурно-ориентированного подхода – введи данные, в зависимости от варьируемых параметров выбери ту или иную вычислительную схему, выполни цепочку процедур и, при необходимости, повтори ее нужное число раз, выведи результаты. Программа такого рода четко определяла последовательность действий, практически, не требующих вмешательства человека, и ее оставалось только запустить.

Однако далеко не все программы могут быть построены по такой схеме. Простейшим примером является обычный текстовый редактор, в котором выполняемые программой операции зависят от взаимодействия с пользователем и включают довольно широкий набор исполнительных процедур, реагирующих на возникающие события.

Другим примером событийно-ориентированной программы является сама операционная система, которая заранее не знает, какие приложения ей придется выполнять, какие ресурсы для этого могут потребоваться как при запуске приложения, так и в процессе его функционирования, какие действия предпримет пользователь и т.д. и т.п. В условиях такой неопределенности операционная среда организована как система массового обслуживания. Она состоит из группы исполнителей, выполняющих определенного рода заявки, асинхронно

поступающие как от аппаратуры, так и от системных и прикладных программ. Возникающие заявки выстраиваются в очереди к соответствующим исполнителям и, по мере возможности, ими обслуживаются по той или иной схеме предпочтения. Во время обработки текущей заявки может произойти новое событие. Тогда работу исполнителя временно приостановят, поставят поступивший заказ в соответствующую очередь, после чего управление возвращается прерванной программе. Если очереди к исполнителям оказываются пустыми, то в операционной системе организуется бесконечный цикл ожидания, прерываемый ближайшим событием. В работе большого количества компонентов ОС важную роль играют сообщения, которыми обмениваются взаимодействующие модули. Количество таких сообщений достигает нескольких сотен.

Описанные выше принципы событийно-ориентированного функционирования ОС заложены в операционных средах типа Windows. Более того, свою архитектуру Windows **требует повторять и в прикладных программах. Это содействует унификации пользовательского интерфейса CUA (Common User Access), разработанного в рамках промышленного стандарта фирмы IBM на архитектуру прикладных программ SAA (System Application Architecture).**

Как правило, любое Windows-приложение должно иметь главное окно стандартного образца, содержащее довольно много управляющих элементов – системное меню для прекращения работы приложения и переключения на другое задание, главное меню, кнопки для свертывания и развертывания окна, специальную рамку с возможностью изменения ее размеров и положения на экране, линейки прокрутки и т.п. Создание такого оформления с использованием программных средств низкого уровня, т.е. на базе системных функций GUI (Graphic User Interface), требует знания многочисленных деталей и умения работать с непривычными понятиями типа контекст устройства, сообщения Windows, ресурсы приложения, обработка особых ситуаций и т.п.

Снижению трудоемкости процесса создания Windows-приложений в значительной мере содействовали разработки фирмой Borland библиотеки OWL (Object Windows Library) и пользующейся существенно более мощной финансовой поддержкой библиотеки классов MFC (Microsoft Foundation Class Library). В этих библиотеках накапливались как объекты интерфейсного типа, так и классы объектов, упрощающие реализацию типовых процедур обработки данных (графика, работа со списками и таблицами, редактирование текстов и т.п.). Несмотря на очевидный прогресс объектно-ориентированного подхода, традиционная работа с постоянно возрастающим количеством объектов и порождающих их классов требует знания многочисленных деталей – иерархии семейств, параметров и характеристик соответствующих методов и т.п. И как только объем такой информации превышает критическую массу, рейтинг соответствующего программного продукта идет на убыль.

Наиболее изящное решение указанных выше проблем, обеспечивающее минимальные затраты на создание Windows-приложений, предлагает технология визуального программирования. Визуальная среда принимает на себя все заботы по оформлению типовых окон – главное окно прикладной программы, диалоговые окна ввода строк и вывода стандартных сообщений. Она предоставляет программисту возможность без каких-либо хлопот встроить в приложение тот или иной объект, приписать ему необходимые свойства, сформировать схему поведения объекта в случае возникновения определенных событий. Прилагательное «визуальный» при этом означает, что основное большинство процедур по заготовке объектов и заданию их свойств выполняются по принципу «делаю то, что вижу, и соглашаюсь на типовые назначения по умолчанию». Ни одного оператора в своем приложении для этих целей писать не приходится. Указав элемент в палитре компонентов, вы можете с помощью мыши отбуксировать объект на нужное место и изменить его размеры. Установка свойств объекта не сложнее выбора строки меню или набора нужного значения в поле ввода. Все заботы по отображению установленных объектов и управлению их поведением во время работы приложения система берет на себя.

Впервые в законченном виде визуальный подход к созданию Windows-приложений был реализован фирмой Microsoft в системе Visual Basic, появившейся в 1991 г. и мгновенно завоевавшей симпатии программистов. В течение двух следующих лет Visual Basic неизменно входил в тройку лучших программных продуктов года. Исполняемый модуль, изготовленный в среде Visual Basic, поражает своими миниатюрными размерами, однако он может работать только в связке с интерпретирующей библиотекой VBRUNxx.DLL (символы xx соответствуют номеру версии - 10, 20, 30, 40). Поэтому размер оперативной памяти, необходимой для запуска такого приложения в десятки раз превышает длину exe-файла. Интерпретация Basic-программы существенно (в 5-10 раз) замедляет работу приложения по сравнению с аналогичными exe-файлами, изготовленными в системах компилирующего типа. В последних версиях этого продукта, появившихся совсем недавно (MS VB-5.0 и MS VB-6.0), технология интерпретации полностью заменена довольно эффективными компиляторами. Библиотеки инструментов VB-систем (VBX- и OCX-компоненты) активно расширяются, на их базе появляются новые профессиональные пакеты и наборы инструментальных средств.

Существенным плюсом ранних версий Visual Basic является неприязнательность к машинным ресурсам. Они могут функционировать на компьютерах устаревших моделей (типа IBM PC/386) с оперативной памятью 4 Мбайт под управлением Windows 3.x.

Окрыленная успехом любимого программного продукта Билла Гейтса (в свое время его карьера начиналась с Бейсик-интерпретатора для 8-разрядных

бытовых компьютеров), фирма Microsoft попыталась завоевать рынок более профессиональных программистов в своей следующей разработке – Visual C++. В этой системе в очередной раз проявилась гигантомания, характерная для мирового лидера, не всегда обращающего внимание на удобства и возможности рядовых пользователей. Размеры первых инсталляционных версий превышали 100 Мбайт, а для их раскрутки требовалось порядка 150-250 Мбайт на винчестере. Для нормального функционирования таких систем необходимы компьютеры не ниже Pentium с оперативной памятью от 16 Мбайт и выше. Да и визуальность в новой системе несколько отошла на второй план.

Одну из наиболее удобных практических систем визуального программирования разработали сотрудники компании Borland International, выпустившие на рынок уже четвертую версию Delphi, **входным языком в которой является объектно-ориентированный Object Pascal**. По большинству показателей система Delphi значительно опередила разработки ²старшего брата² и последние 2-3 г. неизменно называется одним из лучших программных продуктов. На базе версии Delphi-2 сотрудникам фирмы Borland удалось разработать систему программирования C++ Builder 1.0, которая принимает модули, написанные как на C++, так и на Паскале. Совсем недавно на рынке появилась следующая модификация – C++ Builder 3.0, которая наряду с родной библиотекой компонентов VCL (Visual Component Library) позволяет подключать классы и из библиотеки MFC.

При ограниченных ресурсах (оперативная память 8 Мбайт, операционная система типа Windows 3.x) учебный процесс может быть построен на базе визуальной среды Delphi-1.0. **Более современные версии могут эксплуатироваться** только под управлением операционной системы не ниже Windows-95 и весьма положительно реагируют на оперативную память от 16 Мбайт и выше.

Визуальные среды не только упрощают процесс создания современного пользовательского интерфейса. Среди компонент, подключенных к штатному составу библиотек, можно обнаружить управляющие элементы для создания средств телекоммуникации, использующих различные протоколы обмена данными, диалоговые компоненты для управления файловой системой, средства настройки принтеров и печати различного рода отчетов, графические объекты, обеспечивающие выполнение базовых процедур растровой и векторной графики.

Визуальный подход к проектированию типовых элементов интерфейса и обработки данных не мог не оказать влияния на разработчиков программного обеспечения для обширной прикладной сферы, связанной с СУБД. На рынке программных продуктов уже появились визуальные версии FoxPro (Visual FoxPro 3.0 – 6.0) и dBASE (версия dBASE 5.5). В системе Microsoft Access язык Visual Basic является средством программирования прикладных про-

грамм, работающих с базами данных. В состав Delphi и C++ Builder включен довольно мощный инструментальный набор объектов для работы с наиболее распространенными базами данных (dBASE, Paradox, InterBase, Oracle, SysBase, Informix, MS SQL-Server). Этот набор активно пополняется новыми средствами обработки данных, которые в массовом порядке появляются на различных серверах сети Internet.

Новая вычислительная техника в лице одно- и двухпроцессорных систем на базе процессоров Pentium-II, операционные системы типа Windows-98 и Windows-NT постепенно вытесняют маломощные устаревшие модели и ограниченную DOS, так и не справившуюся с проблемой эффективного использования большой оперативной памяти. И не считаться с этим наступлением нельзя. Единственная возможность шагнуть в ногу со временем – начинать как можно раньше знакомить учащихся средних и высших учебных заведений с технологией визуального программирования. Выбор алгоритмического языка и соответствующего программного продукта в данном случае принципиальной роли не играет – функциональные возможности упомянутых выше систем на стадии начального обучения более или менее одинаковы. Некоторое предпочтение (может быть, здесь имеет место субъективизм автора) следовало бы отдать программным продуктам фирмы Borland. Об одном из аргументов в их пользу свидетельствует возможность очень простого перевода программ из DOS в так называемые консольные приложения Delphi или Builder'a. Получающиеся при этом программы почти не отличаются по оформлению от MS-DOS, однако они используют 32-разрядную адресацию и могут работать с любым объемом оперативной памяти.

В заключение приведем далеко не полный список литературы, которая сейчас появляется достаточно своевременно, но по ряду причин экономического характера отсутствует в библиотеках наших вузов.

Visual Basic

1. Арушанов Х.Р. Visual Basic 3.0, Visual Basic 4.0 для Windows. - М.: АБФ, 1995. - 368 с
2. Браун К. Введение в Visual BASIC для программистов. - М.: Мир, 1993. - 416 с
3. Бунин Э. Excel Visual Basic для разработки приложений. - М.: БИНОМ, 1997.
4. Витенко О., Яковлев В. Visual Basic в бюро. - К.: BHV, 1997. - 334 с
5. Комягин В.Б. Программирование в Excel 5 и Excel 7 на языке Visual Basic. - М.: Радио и связь, 1996. - 320 с
6. Курата Д. Создание объектов в Microsoft Visual Basic 4.0. - М.: СК Пресс, 1997. - 442 с

7. Макашарипов С. Программирование баз данных на Visual Basic 5 в примерах. - СПб.: Питер, 1997. -256 с
8. Маккелви М. Visual Basic 4. - М.: БИНОМ, 1997. -572 с
9. Нельсон Р. Running Visual Basic 3 for Windows. - М.: изд. Отдел «Русская редакция», 1995. - 384 с
10. Орвис У. Visual Basic for Applications на примерах. - М.: БИНОМ, 1995. - 512 с
11. Потопкин А.В. Основы Visual Basic для пакета Microsoft Office. - М.: ЭКОМ, 1995. - 254 с
12. Райманс Х.-Г. Введение в курс Visual Basic. - К.: BHV, 1993. - 272 с
13. Санна П. и др. Visual Basic для приложений (версия 5) в подлиннике. - СПб.: BHV, 1997. - 704 с
14. Скотт Д.Ф. Разработка прикладных систем на Visual Basic для Windows. - М.: Исланд, 1994. - 480 с
15. Скотт Р.П. Учимся мастерству Microsoft Visual Basic 4.0. - М.: СК Пресс, 1997. - 352 с
16. Хиллер С. Microsoft Visual Basic, Scripting Edition в действии. - СПб.: Питер, 1997. - 438 с
17. Шмидт В. Microsoft Visual Basic 5.0. - М.: АБФ, 1997. - 687 с

Delphi

1. Возневич Э. Освой самостоятельно Delphi : Полное руководство для самостоятельного обучения. - М.: БИНОМ, 1996. - 736 с
2. Дантемманн Д., Мишел Д., Тейлор Д. Программирование в среде Delphi. - К.: DiaSoft Ltd., 1995. - 608 с
3. Дарахвелидзе П., Марков Е. Delphi - среда визуального программирования. - СПб.: BHV, 1996. - 352 с
4. Епанешников А.М., Епанешников В.А. Программирование в среде Delphi 2. В 4-х частях. - М.: Диалог-МИФИ, Часть 1. Описание среды. 1997. - 235 с Часть 2. Язык Object Pascal 9.0. 1997, - 320 с
5. Калверт Ч. Delphi 2. Энциклопедия пользователя. - К.: ДИАСофт Лтд., 1996. - 736 с
6. Кетков Ю.Л., Лебедев К.В., Петрова Е.В., Уварова Е.П. Технология визуального программирования в среде Delphi. - Н. Новгород : изд. ННГУ, 1997. - 138 с
7. Конопка Р. Создание оригинальных компонент в среде Delphi. - К.: DiaSoft Ltd., 1996. - 512 с
8. Матчо Д., Фолкнер Д.Р. Delphi. - М.: БИНОМ, 1995. - 464 с
9. Миллер Т., Пауэлл Д. Специальное использование Delphi 3. - М.,К.: Диалектика, 1997. - 768 с

10. Мэтчо Д. Delphi 2. Руководство для профессионалов. - СПб.: BHV, 1997. - 784 с
11. Орлик С. Секреты Delphi на примерах. - М.: БИНОМ, 1996. - 316 с
12. Оузьер Д., Гробман С., Батсон С. Освой самостоятельно Delphi 2. - М.: БИНОМ, 1997. - 614 с
13. Рубенкинг Н. Программирование в среде Delphi для «чайников». - К.: Диалектика, 1996. - 304 с
14. Сван Т. Основы программирования в Delphi для Windows 95. - К.: Диалектика, 1996. - 480 с
15. Сван Т. Секреты 32-разрядного программирования в Delphi. - К.: Диалектика, 1997. - 472 с
16. Сурков К., Сурков Д., Вальвачев А. Программирование в среде Delphi 2.0. - Минск: Попурри, 1997. - 640 с
17. Тейлор Д., Мишель Дж., и др. Delphi 3: Библиотека программиста. - СПб: Питер, 1998. - 560 с
18. Туррот П., Брент Г., Багдазиан Р., Тендон С. Супербиблия Delphi 3. - К.: Диасофт, 1997. - 848 с
19. Фаронов В.В. Delphi 3.0. Учебный курс. - М.: Нолидж, 1998. - 400 с
20. Федоров А. Создание Windows-приложений в среде Delphi. - М.: КомпьютерПресс, 1995. - 287 с
21. Федоров А.Г. Delphi 2.0 для всех. 2-е изд., перераб. и доп. - М.: КомпьютерПресс, 1997. - 464 с
22. Хендерсон К. Руководство разработчика баз данных в Delphi. - К.: Диалектика, 1996. - 544 с
23. Хендерсон К. Delphi 3. Системы Клиент/Сервер. - К.: Диалектика, 1997. - 736 с
24. Шумаков В.П. Delphi 3 и создание приложений баз данных. - М.:Нолидж, 1998. - 704 с
25. Энго Ф. Как программировать на Delphi 3. - К.: Диасофт, 1997. - 320 с
26. Энго Ф. Самоучитель Delphi 3. - К.: Диасофт, 1998. - 320 с

Borland C++ Builder

1. Елманова Н.З., Кошель С.П. Введение в Borland C++ Builder. - М.: Диалог-МИФИ, 1997. - 272 с
2. Елманова Н.З. Borland C++ Builder 3.0. Архитектура «клиент/сервер». Многозвенные системы и Internet-приложения. - М.: ДИАЛОГ-МИФИ, 1998. - 240 с
3. Калверт Ч. Borland C++ Builder. Энциклопедия пользователя. - К.: Диасофт Лтд., 1998. - 848 с

4. Рейсдорф К., Хендерсон К. **Borland C++ Builder. Освой самостоятельно.** - М.: БИНОМ, 1998. - 704 с
5. Шамис В.А. **Borland C++ Builder. Программирование на C++ без проблем.** - М.: Нолидж, 1997. - 266 с

Microsoft Visual C++

1. Биллиг В.А., Мусикаев И.Х. **Visual C++, 4 версия: Книга для программистов.** - М.: Русская редакция, 1996. - 328 с
2. Мешков А., Тихомиров Ю. **Visual C++ и MFC. Программирование для Windows NT и Windows 95. В 3-х томах.** -СПб.: ВHV, 1997
3. Тосс В. **Visual C++ 5. Энциклопедия пользователя.** - К.:Диасофт, 1998. - 704 с
4. Фролов А.В., Фролов Г.В. **Microsoft Visual C++ и MFC. Программирование для Windows 95 и Windows NT.** - М.: ДИАЛОГ-МИФИ, 1996. - 288 с
5. Фролов А.В., Фролов Г.В. **Разработка приложений для Internet. Visual C++ и MFC. (В среде Windows 95 и Windows NT).**- М.: ДИАЛОГ-МИФИ, 1997. - 286 с
6. **Microsoft Visual C++ 4.** - М.: ЭНТРОП, К.: Век+, 1997. - 702 с

Кетков Ю.Л.

Межвуз.сб. Новые информационные технологии и развитие образования. Вып.286,
ВГАВТ, Н. Новгород, 1999 г.

Об одном подходе к подготовке профессиональных программистов

Статья носит дискуссионный характер и предлагает обсудить вопрос о технологии преподавания информатики в высших учебных заведениях. Заранее оговоримся, что мы не имеем ввиду сиюминутные курсы уровня пользователя ПК, имеющие целью привить ограниченные навыки работы с компьютером и дать минимальный набор рекомендаций по применению того или иного популярного программного продукта. Речь идет о подготовке специалистов, способных создавать собственные прикладные и системные программы. Ибо ориентация только на использование кем-то разработанных программных и аппаратных средств вычислительной техники обрекает нас на отставание в области новых информационных технологий.

Основная идея заключается в четком выделении в учебных курсах типа «Методы программирования» или «Алгоритмические языки» двух уровней. Первый из них, – базовый, – строится на параллельном изложении основ трех алгоритмических языков – Бейсика, Паскаля и Си. Второй уровень предполагает достаточно глубокое изучение возможностей Ассемблера IBM-совместимых ПК и одной из современных сред визуального программирования – Visual Basic, Visual C++ (альтернатива – Borland C++ Builder) или Delphi.

Ни для кого не является секретом, что выпускники средних школ в той или иной мере обладают практическими навыками работы с достаточно примитивными подмножествами алгоритмических языков Бейсик или Паскаль. В силу технической оснащенности наших школ большинство учащихся (порядка 60%) знакомятся с весьма ограниченной версией языка Бейсик для морально и физически устаревших ПК типа Корвет или КУВТ. Гораздо меньший контингент (порядка 15-20%), к которому относятся преимущественно выпускники городских и специализированных школ, владеют навыками работы в среде Turbo Pascal. Однако в любом случае такие разделы как работа с файлами, машинная графика, использование отладочных средств и обработка прерываний, как правило, остаются вне досягаемости школьных курсов.

Совершенно очевидно, что школа может познакомить учащихся лишь с ограниченным инструментарием того или иного алгоритмического языка. Но она не в состоянии научить своих выпускников умению использовать этот инструмент для решения широкого класса практических задач. Слишком мал у школьников запас знаний по методам вычислительной математики. Невелик у них и опыт практической работы с компьютером. Ощущается явный недостаток знаний возможностей библиотечных функций и операционных систем, умения применять их на практике. В силу ограниченности времени и средств

вряд ли стоит относить эти претензии к средней школе. Да и не должна школа готовить всех своих выпускников к карьере профессионального программиста.

Министерские программы курсов, связанных с преподаванием информатики, к счастью, не ограничивают вузы в выборе алгоритмических языков. По сложившейся в нашей стране традиции естественные факультеты государственных университетов тяготеют к Паскалю, в технических вузах предпочитают изучение Си, а на факультетах экономического и социально-политического профиля преобладают курсы, ориентированные на Бейсик и его дальнейшее развитие в лице VBA (Visual Basic for Applications).

Сложившееся у нас распределение не очень отвечает запросам современного рынка труда. Так, например, в США основная масса прикладных программистов работает в среде Visual Basic. По свидетельству [1] из примерно 5 млн. программистов в мире около 2 млн. создают программные продукты на базе VB или VBA. Однако наиболее квалифицированную работу и, соответственно, существенно более высокую зарплату могут найти специалисты, владеющие техникой программирования в средах типа Visual C++. Не следует сбрасывать со счетов и язык Паскаль, появление которого было обусловлено, в первую очередь, проблемой обучения качественному программированию. На этом языке в свое время была написана основная часть операционной системы Windows. Средствами языка Object Pascal были реализованы такие профессиональные среды визуального программирования как Delphi и Borland C++ Builder.

Многие вузы, обеспокоенные судьбой своих выпускников, пытаются отслеживать тенденции современного спроса. Так, например, успехи Интернет побудили высшие учебные заведения Великобритании перестроить программы обучения информатики, ориентируя их на язык межплатформенного общения Java.

На наш взгляд, тенденция к моноязыковому подходу в подготовке профессиональных программистов в принципе неверна. Во-первых, естественный консерватизм в разработке и реализации новых учебных программ не в состоянии своевременно отслеживать быстро прогрессирующую сферу разработки программных продуктов. Во-вторых, совершенствование алгоритмических языков и современных систем программирования сопровождается четко выраженной унификацией семантических форм и синтаксических конструкций. Достаточно развитые алгоритмические языки универсального назначения обладают сегодня очень сходным набором функциональных возможностей. Их параллельное изучение с использованием явных или близких аналогий позволит готовить системных и прикладных программистов с более широким кругозором, способных легко ориентироваться в огромном разнообразии программных средств, написанных на разных алгоритмических языках. Наконец, нам представляется достаточно важным воспитать умение творчески адаптировать

программы, написанные на другом языке. Несмотря на то, что такой перевод может быть выполнен автоматически, его качество в большинстве случаев оставляет желать лучшего.

Компьютерный полиглот, на наш взгляд, сродни высоко образованному человеку, свободно владеющему двумя или более иностранными языками. Известны факты, когда знатоки десятка языков затрачивают на освоение нового языка всего 2-3 месяца.

Фундамент многоязыкового образования профессиональных программистов может быть построен на параллельном изучении трех достаточно разных по своему синтаксису языков – Бейсика, Паскаля и Си. В пользу такого выбора имеются следующие основания. Во-первых, основы Бейсика или Паскаля закладываются программами школьного образования. Во-вторых, большинство наиболее серьезных приложений за последние 30 лет написаны на той или иной версии Си. Кстати, язык Java, царящий ныне в Интернет, фактически, является подмножеством языка C++. Наконец, для каждого из этих языков существуют достаточно близкие по своей форме среды программирования, не предъявляющие сколь-нибудь серьезных требований к конфигурации ПК. Мы имеем ввиду интерпретатор QBasic, входящий в состав MS-DOS, и практически идентичные по форме интегрированные среды Borland C++ (версия 3.1) и Turbo Pascal (версии 6.0 или 7.0).

В качестве практического пособия, ориентированного на формирование предложенного фундамента, мы подготовили сборник задач, включающий более 130 готовых к исполнению программ различной сложности. Сборник развивает идеи параллельного знакомства с разными алгоритмическими языками, заложенными нами в [2]. Более 100 задач в сборнике представлены тремя или более версиями программ, демонстрирующих специфику того или иного языка.

Сборник состоит из 10 глав, охватывающих следующие разделы:

- Работа с числовыми данными (внешнее и внутреннее представление числовых данных, ввод и вывод числовой информации);
- Обработка текстовой информации (символьные данные и их внутреннее представление, ввод и вывод текстовой информации, обработка фрагментов строк, сравнение и сортировка текстовых данных, управление цветом в текстовом режиме)
- Работа с массивами (объявление и инициализация массивов, статические и динамические массивы, массивы в качестве параметров процедур и функций, сортировка больших массивов, поиск);
- Рекурсивные функции и процедуры;
- Подпрограммы и функции (оформление и вызов программных единиц, параметры подпрограмм, локальные и глобальные данные);

- Работа с дисковыми файлами (строковые, записеориентированные и двоичные файлы);
- Машинная графика (графические возможности систем программирования, инициализация графического режима, определение области графического вывода и выбор системы координат, управление цветом, работа с отдельными точками и растровыми изображениями, отрезки прямых и прямоугольники, окружности, эллипсы и дуги, закрашивание и заполнение замкнутых областей, заливка площадных фигур «прозрачными» шаблонами, текстовые сообщения в графическом режиме);
- Работа с календарными датами (вычисление юлианских дат);
- Использование системных функций (управление мышью, красивые окна в текстовом режиме)

Каждому разделу предшествует краткий, но достаточно полный обзор соответствующих программных средств, присущих избранным алгоритмическим языкам. Большинство задач сопровождается практическими советами по построению тех или иных компонентов программы с учетом специфики каждого языка. При формировании даже простейших программ особое внимание уделяется их структуре, постоянно подчеркивается возможность выделения и использования процедур (подпрограмм) и функций.

Литература

1. Бобровский С. Программирование на языке QBasic для школьников и студентов. – М.: «ДЕСС КОМ», 2000. – 207 с.
2. Кетков Ю.Л., Кетков А.Ю., Шапошников Д.Е. Персональный компьютер: Школьная энциклопедия. – М.: Большая Российская Энциклопедия, 1998. – 440 с.

Кетков Ю.Л., Кетков А.Ю.

В. сб. ВГАВТ, Н.Новгород, 2000 г.

О некоторых пионерских работах на первых ЭВМ

Машина ГИФТИ и схема Горнера

Первый выпуск математиков-вычислителей в Горьковском университете состоялся в 1957 г. благодаря поддержке со стороны д.ф.-м.н. А.А. Ляпунова в то время зав. отделом программирования Отделения прикладной математики Математического института им. В.А. Стеклова АН СССР. Он же одновременно был профессором мехмата МГУ. Весь учебный 1956–1957 г. шесть студентов физмата ГГУ провели в Москве, т.к. компьютеров в Горьковской области, включая и всемогущий ВНИИЭФ (г. Саров), еще не было. Мы посещали в МГУ лекции по программированию, которые начал читать молодой аспирант А.П.Ершов, проходили годичную стажировку на первых цифровых ЭВМ М-2 (в лаборатории И.С. Брука), БЭСМ (в ИТМ и ВТ), Стрела-1 (в ОПМ МИ АН СССР), выполняли дипломные работы. Почти весь наш выпуск был по распределению направлен на работу в Горьковский Исследовательский Физико-технический Институт (ГИФТИ при ГГУ), где в это время создавался первый в Горьковской области вычислительный центр (ВЦ ГИФТИ).

Поначалу главным средством вычислений была самая мощная аналоговая машина МН-8, которая обеспечивала решение систем дифференциальных уравнений высокого порядка. Она было очень важным подспорьем для специалистов по теории колебаний, которые занимались исследованием процессов управления и устойчивостью работы ядерных реакторов, проектировавшихся для первых АЭС и атомных подводных лодок. Одновременно заканчивалась настройка небольшой цифровой ЭВМ, получившей название «Машины ГИФТИ». Она разрабатывалась группой выпускников кафедры теории колебаний радиофизического факультета ГГУ, долгое время возглавляемой академиком А.А. Андроновым. В свое время А.А. Андронов был членом межведомственных комиссий, проверявших работу ряда академических подразделений, связанных с проектированием первых ЭВМ в нашей стране. Он здраво оценил перспективность нового направления и всячески содействовал развитию этой тематики на своей кафедре.

В отличие от ЭВМ «Стрела-1», занимавшей в ОПМ двухэтажное здание и его подвальные помещения, с обслуживающим персоналом более 200 человек машина ГИФТИ спокойно размещалась в небольшой комнате и потребляла около 15 квт электроэнергии. Она была машиной последовательного действия с оперативной памятью в 2016 тридцатидвухразрядных ячеек, расположенных на магнитном барабане. На этом же барабане были реализованы сверхбыстрые

рециркуляционные регистры, позволившие довести скорость работы арифметического устройства до 6000 сложений в сек. Операции деления-умножения, естественно, выполнялись примерно в 30 раз медленнее. Общая производительность машины ГИФТИ сдерживалась медленной оперативной памятью и в среднем составляла около 100 операций в сек.

Трое первых программистов, включая автора, были нацелены на создание базового программного обеспечения – тестовое хозяйство, библиотеки стандартных функций, простейшие методы решения алгебраических и дифференциальных уравнений. Скорость вычислений была для нас главным критерием качества программ. Особенно удручала небольшая скорость выполнения операций деления-умножения. Поэтому первой пионерской работой, позволившей на 25% ускорить работу подпрограмм вычисления элементарных функций, была борьба с традиционной схемой Горнера. Как правило, для вычисления значений элементарных функций использовались полиномы 8-10 степени, полученные тем или иным способом (разложение в ряд Тейлора, непрерывные дроби, снижение старших степеней за счет полиномов Чебышева и т.п.). Схема Горнера позволяла вычислить значение полинома n -ой степени в заданной точке за n операций умножения и n операций сложения:

$$P_n(x) = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \dots + a_1 \cdot x + a_0$$
$$P_n(x) = (\dots((a_n \cdot x + a_{n-1}) \cdot x + \dots + a_1) \cdot x) + a_0$$

Её оптимальность сомнений не вызывала: чтобы задействовать $n+1$ независимых коэффициентов a_i без n сложений никак нельзя было обойтись. А по поводу умножений сомнения возникали – обязательно ли было каждым умножением повышать степень промежуточного результата только на 1. Ведь из алгебры известно, что полином можно представить в виде произведения скобок, содержащих выражения вида $(x - x_j)$, где x_j – корни полинома. В том случае, когда один из корней оказывался комплексным $(x_j = c + i \cdot d)$, для него всегда находился сопряженный $(c - i \cdot d)$, и произведение двух парных скобок заменялось вещественным полиномом 2-й степени. Дело было за малым – надо найти все корни полинома (и всё это ради только того, чтобы вычислить значение полинома при заданном аргументе x). Из той же алгебры следовало, что полиномы выше 4-й степени в радикалах неразрешимы. Следовательно, надо было находить корни приближенным способом и исследовать распространение ошибок первого шага на общую погрешность вычисления полинома. По сравнению с этим схема Горнера казалась блестящей находкой (и даже сегодня многие в этом уверены). Гораздо более перспективным казался путь, подсказанный публикацией статьи американского математика Дж. Тодда [1], где со ссылкой на работу [2] была приведена частная схема вычисления полинома

6-й степени с повышением степени промежуточного результата на 2 при каждом умножении. Эта схема требовала предварительного пересчета исходных коэффициентов $(a_6, a_5, a_4, a_3, a_2, a_1, a_0)$ в 6 других также независимых коэффициентов (b_6, b_5, \dots, b_0) и позволяла сэкономить 2 умножения. Все было бы хорошо, если бы на стадии пересчета при некоторых комбинациях исходных коэффициентов не получались бы квадратные уравнения, не имевшие вещественных корней. А появление комплексных коэффициентов b_j сводило на нет все преимущества схемы Тодда.

В 1957 г. мне удалось придумать более универсальную схему, напоминающую американский аналог, но для повышения степени промежуточного полинома на 2 она требовала через шаг то одного умножения, то двух. Зато пересчет исходных коэффициентов никогда не приводил к комплексным b_j . По сравнению со схемой Горнера экономия в количестве умножений составила 25%. Предварительные затраты на пересчет исходных коэффициентов во внимание принимать не надо, т.к. эти затраты разовые, а все последующие обращения к подпрограмме вычисления стандартной функции выполнялись по укороченной схеме. Свою первую апробацию новые стандартные программы прошли на машине ГИФТИ. Впервые эта схема была опубликована в 1958 г. в относительно мало известном для программистов журнале «Известия вузов. Радиофизика» [3], а позднее по рекомендации Л.А. Люстерника она была включена в один из сборников «Справочной Математической Библиотеки» (СМБ, [4]).

УИС РГМ – прообраз первых электронных таблиц на ЭВМ типа М-20

8 марта 1961 г. в ВЦ ГИФТИ была введена в эксплуатацию одна из лучших ЭВМ первого поколения М-20. После этого машина ГИФТИ была передана кафедре теории колебаний радиофакультета в качестве средства расширения материальной базы учебного процесса. Перед группой программирования, переключившейся на М-20 и насчитывающей уже порядка 15 человек, были поставлены задачи поиска внешних заказчиков и повышения квалификации в области решения разнообразных прикладных задач. Довольно стабильным поставщиком задач, связанных с управлением атомными реакторами, была специализированная лаборатория ГИФТИ, возглавляемая Н.А. Железцовым. В силу специфики тематики она выступала как организация «п/я 88». Затем к нам потянулись радиофизики, специалисты по строительной механике, конструкторы новых радиолокационных станций. В начале 1963 г. к руководству ВЦ обратился главный инженер ЦКБ «Волгобалтсудопроект» А.А. Брайловский с просьбой помочь в выполнении плазовых расчетов для сухогруза типа

«река-море» (проект 1829) по схеме, разработанной в НИИ Технологии Машиностроения (НИИТМ, Ростов-на-Дону). Автором этой схемы был главный конструктор СКБ НИИТМ Д.С. Китаинов, а сама технология образмеривания всех практических сечений корпуса судна носила название радиусографического метода. Название метода объясняется тем, что все продольные (ватерлинии, батоксы) и поперечные (шпангоуты) сечения проектируемой поверхности формировались из отрезков прямых и дуг окружностей. На большинстве участков судовой поверхности они сопрягались с сохранением значения первой производной на линиях стыков.

Для того чтобы пояснить суть расчетов по радиусографическому методу, опишем некоторые элементы технологии проектирования судов и предварительной подготовки чертежной документации на судостроительных заводах нашей страны. Два крупнейших центра по проектированию надводных и подводных судов располагались в городах Горький и Ленинград. В их составе насчитывалось около десятка специализированных конструкторских бюро, которые в соответствии с техническими заданиями разрабатывали проекты конкретных изделий. Наряду с уточненными тактико-техническими данными корабля, составом и размещением его оборудования основу проекта составлял теоретический чертеж судовой поверхности, выполненный в трех взаимно перпендикулярных плоскостях. Сотрудники КБ составляли такие чертежи в масштабе от 1:25 до 1:100 в зависимости от размера судна. Как правило, на чертеже, поступившем из КБ, были представлены поперечные сечения судна в виде 21 теоретического шпангоута. Продольных сечений (ватерлинии, батоксы) было существенно меньше. Особое внимание уделялось сечениям, характеризующим главные измерения судна (мидель-шпангоут, грузовая ватерлиния и др.). Одна из проекций с поперечными сечениями носа и кормы показана на рис. 1.



Рис.1. Проекция «Корпус»

Теоретический чертеж неминуемо сопровождался большим количеством неточностей, нестыковкой координат одних и тех же точек судовой поверхности на разных сечениях. Ошибка в 1 мм с теоретического чертежа увеличивалась до 10 см на реальной поверхности. Поэтому графические данные теоретического чертежа обязательно проходили согласование в процессе плазово-разметочных работ, выполняемых в специальных (плазовых) цехах судостроительных предприятий. Так, например, плазовый цех Сормовского судостроительного завода представлял собой одноэтажное здание длиной порядка 200 м с идеально выровненным полом. На этом полу несмываемой тушью в натуральную величину воспроизводились все элементы теоретического чертежа с добавлением нужного количества промежуточных сечений. Плавность линий достигалась на глаз с помощью специальных гибких реек («правил», в англоязычной терминологии - «сплайнов»), которые прижимались в опорных точках металлическими грузиками («крысами») в форме небольших утюжков. Линия считалась идеальной, если после снятия нагрузки рейка сохраняла приданную ей форму и отклонялась в опорных точках не более чем на 1-2 мм. Субъективизм в построении плавных линий приводил к тому, что плазовые чертежи одного и того же проекта, выполненные на разных судостроительных заводах, слегка отличались. Поэтому ремонт судна всегда должен был выполняться на том заводе, где корабль был построен.

Основу радиусографического метода составлял так называемый радиусографический ключ – набор небольшого количества продольных линий, которые проходили вдоль корпуса судна и сами были составлены из отрезков прямых и дуг окружностей. Некоторые линии ключа были расположены на поверхности судна. Точки ключевых линий определяли опорные точки на судовой поверхности и координаты центров всех дуг, образующих поперечные сечения. Построение ключа велось обычно от прототипа изделия и определялось пространственным воображением и опытом проектировщика. Д.С. Китаинов виртуозно владел технологией формирования радиусографических ключей, но с большой неохотой передавал свой опыт другим проектировщикам. После того, как радиусографический ключ был сформирован, наступал весьма кропотливый и трудоемкий процесс вычисления координат всех точек сопряжения смежных отрезков и дуг окружностей, радиусов закруглений на каждом из практических сечений. Для этой цели в СКБ НИИТМ было сформировано две библиотеки соответствующих геометрических процедур, каждая из которых содержала порядка 40 типовых геометрических задач и численных схем их решения. Разница между библиотеками заключалась только в форме задания уравнения прямой (через угловой коэффициент или через пару точек). Таким образом, задача построения согласованных плазовых таблиц сводилась к многократному выполнению формируемых цепочек геометрических задач. Но автоматизировать этот процесс сотрудникам НИИТМ никак не удавалось. Оказались несостоятельными и те немногие организации Ростова-на-Дону, которые располагали средствами вычислительной техники. А сотрудникам ВЦ ГИФТИ, получившим начальную подготовку в одном из лучших вычислительных центров страны и успешно осваивавших идеи М.Р. Шура-Буры, заложенные в интерпретирующей системе ИС-2 [6], удалось за год построить несколько версий универсальных интерпретирующих систем радиусографического метода.

Основной идеей построения «матричных систем программирования» послужила мысль об объединении формата команды трехадресной вычислительной машины М-20 и способа задания форматов исходных данных для процедур, включенных в состав библиотеки ИС-2.

π	КОП	A1	A2	A3
-------	-----	----	----	----

- π – признаки модификации адресов (3 бита, одна восьмеричная цифра);
- КОП – код операции (две восьмеричные цифры);
- A1, A2 – адреса первого и второго операндов (по четыре восьмеричные цифры);
- A3 – адрес результата операции (четыре восьмеричные цифры).

π	16	*+1	7501	7610
		Аpar1	Нсп	Аpar2

Обращение к процедуре с номером Нсп.

16 – команда передачи управления на точку входа в ИС-2 (ячейка с адресом 7501);

7610 – ячейка, в которой формируется команда возврата по адресу *+1, т.е. по адресу команды, содержащей вторую строку обращения;

Аpar1, Аpar2 – адреса данных, необходимые для работы вызываемой процедуры.

Абстрактный формат «команды матричного программирования» должен был выглядеть следующим образом:

Аsource	КМО	Аdest
---------	-----	-------

Аsource – адрес (номер) строки в матрице исходных данных

КМО – код матричной операции

Аdest – адрес (номер) строки в матрице результатов

Анализируя состав библиотек с геометрическими построениями, мы обнаружили, что в каждой из задач количество исходных данных не превосходило восьми. Поэтому в качестве длины строки исходных данных была принята цифра 8, а сами исходные значения адресовались одной восьмеричной цифрой – от 0 до 7. Полный адрес любой компоненты исходных данных идентифицировался четверкой восьмеричных цифр, первые три из которых соответствовали номеру строки в «матричной программе», а младшая определяла порядковый номер аргумента в соответствующем векторе исходных данных. С тем, что в ряде процедур количество исходных данных было заведомо меньше восьми, мы смирились, т.к. единообразие в адресации для нас было важнее. Примерно так же обстояло и дело со стандартизацией формата строк матрицы результатов – во всех задачах их количество не превышало трех.

Было совершенно очевидно, что набор кодов матричных программ не исчерпывался только задачами геометрических построений. Нужно было подключить логику (условные и безусловные переходы), команды организации циклов, команды ввода и вывода исходных данных.

Матричная программа для расчета поперечных или продольных сечений представляла собой циклически повторяющийся список геометрических построений, разбитый на два раздела. В первом разделе для каждого значения циклически повторяющейся координаты (x – для шпангоутов, y – для батоксов, z – для ватерлиний) по соответствующим фрагментам ключевых линий определялись координаты опорных точек и центров дуг, образующих текущее сече-

ние. Полученные результаты засылались в соответствующие клетки матрицы исходных данных второго раздела, где была закодирована последовательность геометрически процедур, вычислявших геометрические параметры продольного или поперечного сечения. После того, как аналитика сечения была рассчитана, с заданным шагом вычислялись плазовые координаты точек, расположенных на данном сечении. На небольшом примере поперечного сечения (рис. 2) продемонстрируем технологию задания соответствующего фрагмента матричной программы.

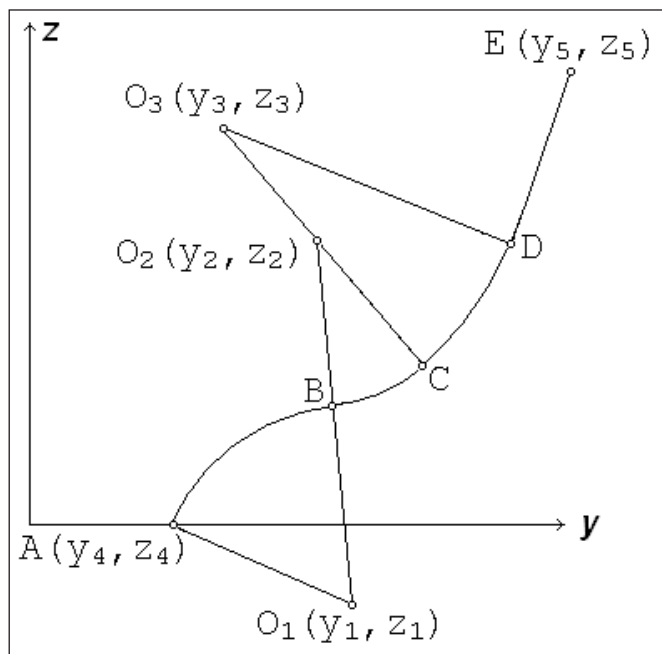
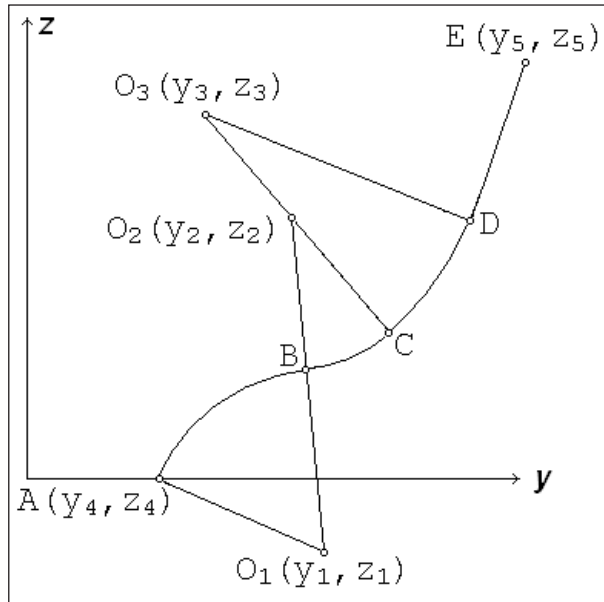


Рис. 2. Схема шпации. Центры окружностей O_1, O_2, O_3 расположены на продольных линиях радиусографического ключа. Точки А и Е находятся на ключевых линиях, проложенных по поверхности судна. Координаты точек сопряжения В, С и D должны быть вычислены по соответствующей цепочке геометрических задач



В табл.1 приведена запись последовательности геометрических построений и соответствующих исходных данных, которые готовил проектировщик формы судовой поверхности. С целью сокращения объема таблицы данные по расчету ключевых параметров уже занесены в соответствующие позиции в виде конкретных числовых значений.

Таблица 1

№ стр.	Исходные данные								Нсп	Выдача на печать			Дальнейшая адресовка		
	0	1	2	3	4	5	6	7		4	2	1	4	2	1
1	20	-10	15	0					12			R_1			31, 125
2	20	-10	22	30					12			n_1			30
3	n_1	R_1							41			R_2			51, 135
4	22	30	12	49					12			n_2			50
5	n_2	R_2							40			R_3			65
6	60	72		12	49	R_3			16	z_D	y_D		73	72	
7	60	72	y_D	z_D					02	b	k				
10	20	-10	22	30					02	b_1	k_1		121	120	
11	19	49	22	30					02	b_2	k_2		131	130	
12	k_1	b_1		20	-10	R_1	-1		70	z_B	y_B				
13	k_2	b_2		22	30	R_2	+1		70	z_C	y_C				

Перенос содержимого этой таблицы на бланки матричных программ осуществляется почти механически. Эквивалент табл.1 в формате, пригодном для ввода в ЭВМ, имеет следующий вид (табл.2).

Таблица 2

π	Код	Адрес	Рассылка	Комментарий
1	12	0010		Вычисление и печать R_1
1			0031	Засылка R_1
1			0125	« – « – «
1	12	0020		Вычисление и печать n_1
1			0030	Засылка n_1
1	41	0030		Вычисление и печать R_2
1			0051	Засылка R_2
1			0135	« – « – «
1	12	0040		Вычисление и печать n_2
1			0050	Засылка n_2
1	40	0050		Вычисление и печать R_3
1			0065	Засылка R_3
3	16	0060		Вычисление и печать z_D, y_D
2			0073	Засылка z_D
1			0072	Засылка y_D
3	02	0070		Вычисление и печать b, k
3	02	0100		Вычисление и печать b_1, k_1
2			0121	Засылка b_1
1			0120	Засылка k_1
3	02	0110		Вычисление и печать b_2, k_2
2			0131	Засылка b_2
1			0130	Засылка k_2
3	70	0120		Вычисление и печать z_B, y_B
3	70	0130		Вычисление и печать z_C, y_C

Первые две версии интерпретирующих систем УИС РГМ и УИС РГВ отличались только наборами библиотечных процедур. Третья версия была пополнена операциями развертки листов судовой обшивки. Первая публикация по технологии автоматизации радиусографического метода [5] носила закрытый характер, т.к. ряд проектов, по которым были выполнены расчеты плазовой документации, относились к оборонной промышленности. Первая открытая публикация появилась два года спустя [7]. Тем не менее, прототип электронных таблиц на отечественных ЭВМ появился примерно за 15 лет до официального представления системы VisiCalc. Цифровая индексация клеток матричной программы была мерой вынужденной, т.к. алфавитно-цифровые устройства подготовки данных в нашей стране еще не производили. Всего в ВЦ ГИФТИ за период с 1963 по 1965 гг. были проведены плазовые расчеты по 19 надводным

судам. Автор причастен к внедрению разработанных комплексов в ряде организаций авиационной и судостроительной промышленности.

Первые линейные сплайны

Попытки визуализации результатов графо-аналитических построений привели нас к использованию первых перьевых плоттеров. Усовершенствование управления их пишущим узлом начинались с того, что сначала перо могло перемещаться на минимальный шаг в одном из четырех направлений. Затем поворот на угол, кратный 90° , был заменен перемещениями под углом, кратным 45° . Наконец, графопостроители начали снабжать интерполяторами, сначала только линейными, а позднее – линейно-круговыми. Это позволило в программе управления плоттером задавать величину перемещения, кратную нескольким минимальным шагам по каждой из координат. В связи с этим возникла задача построения оптимального алгоритма кусочно-линейной аппроксимации произвольного криволинейного контура. И формулировалась-то она предельно просто: как далеко можно было шагнуть вдоль кривой, чтобы соответствующая хорда отклонилась от исходной линии не более чем на заданную величину. Для дуг окружностей существует аналитическая формула, связывающая длину хорды со стрелкой прогиба. А что делать с другими кривыми, например с эллиптическими или параболическими? А если исходная кривая задана таблично? Идентичная задача была актуальной и для станков с числовым программным управлением. Сокращение количества звеньев ломаной, аппроксимирующей обрабатываемый контур, означало уменьшение объема управляющей программы.

Самый простой вариант решения поставленной задачи был очевиден: используя ЭВМ, можно было построить программу, которая постепенно увеличивала длину очередного шага до тех пор, пока стрелка прогиба обеспечивала допустимую точность ϵ . Однако идея тупого перебора казалась мало привлекательной – хотелось использовать более качественные математические методы, позволяющие за один шаг построить звено ломаной почти оптимальной длины. В такой постановке задача становилась вариационной, т.к. границы смежных звеньев оказывались переменными. Её удалось сначала решить для кривых с монотонно меняющейся кривизной при малых значениях допустимой точности ϵ отклонения звеньев ломаной. Идея решения напоминала метод малого параметра, используемый при решении задач устойчивости динамических систем. Величина стрелки прогиба в точке, максимально удаленной от звена ломаной, разлагалась в ряд. В силу малости отбрасывались члены ряда с более высокими степенями ϵ , а затем находились корни уравнения сначала второй степени. Потом аналогичная процедура повторялась для полинома третьей

степени, в котором коэффициент при ε^3 заменялся соответствующим квадратичным полиномом Чебышева, наименее, уклоняющимся от нуля.

Так были получены формулы для квазиоптимального шага, которые численно проверялись на различных кривых. Для одного и того же фрагмента кривой определялось количество $N1$ звеньев ломаной, построенных в сторону уменьшения кривизны, и количество $N2$ звеньев, построенных в обратном направлении. На довольно протяженном участке исходной кривой разность $N1 - N2$ редко превышала 1. Аппроксимация в направлении уменьшения кривизны всегда приводила к звеньям, которые уклонялись от исходной прямой меньше, чем на ε . А в обратном направлении все звенья отклонялись чуть больше, чем на ε . Таким образом можно было установить верхнюю и нижнюю оценки для числа звеньев оптимальной ломаной. Более подробные результаты этого исследования и формулы для вычисления длины шага можно найти в работе [8]. Несколько позже аналогичное исследование было выполнено для кривых 2-го порядка с произвольным значением ε . Его результаты опубликованы в работе [9].

Чтобы оценить результаты проведенных исследований, приведу заключительную фразу из автореферата диссертации [10], защищенной в 1966 г.: «Дальнейшее совершенствование аналитических способов задания криволинейных поверхностей выдвигает ряд новых математических проблем. Одной из них, в частности, является задача кусочно-полиномиальной аппроксимации плоских кривых и поверхностей». Таким образом, в этой работе была практически решена задача построения **оптимальных сплайнов первого порядка** и поставлена задача построения **оптимального набора сплайнов более высокого порядка**. В отечественной литературе это была первая работа в области сплайн-аппроксимации. Да и рубежом первые публикации по сплайнам появились пару лет спустя. Наиболее полная библиография первых зарубежных публикаций, приведенная в книге А.Фокса и М. Пратта «Вычислительная геометрия», ссылается на одну работу 1967 г. (Гревилл), две работы 1968 г. (Безье и Ауджа) и одну работу 1969 г. (Гордон).

Литература

1. Todd J. Communications on Pure and Applied Mathematic, v.8, № 1, 1955 (русский перевод: Дж. Тодд. Мотивы для работы в области численного анализа // «Математическое просвещение», вып.1, 1957)
2. Motzkin T.S. Evaluation of the Polinomials. «Bull. Amer. Math. Society», 1955, v. 61, №2, p.163.
3. Кетков Ю.Л. Об одном способе вычисления полиномов на математических машинах. // Известия вузов. Радиофизика, т.1., № 4, 1958.
4. Люстерник Л.А., Червонскис О.А., Ямпольский А.Р. Математический анализ. Вычисление элементарных функций. М.: Гос. изд. физ. мат. литературы, 1963

5. Кетков Ю.Л. Монография по спецтемаике. Ростов-на-Дону, НИИТМ, 1964, 98 с.
6. Шура-Бура М.Р. Система интерпретации ИС-2// Сб. «Библиотека стандартных программ», Изд-во ЦБТИ, М., 1961.
7. Кетков Ю.Л. Автоматизация проектирования поверхностей корпусов судов при помощи ЭВМ. // В сб. Автоматизация технологического проектирования при помощи электронных вычислительных машин. М.: Машиностроение, 1966. – с.55-72.
8. Кетков Ю.Л. Об оптимальных методах кусочно-линейной аппроксимации. // Известия вузов. Радиофизика, т.9, №6, 1966
9. Кетков Ю.Л. О приближенных методах кусочно-линейной аппроксимации плоских кривых. // Ученые записки. Прикладная математика и кибернетика. Материалы к Всесоюзному межвузовскому симпозиуму по прикладной математике и кибернетике. Горький, 1967, с. 202-211
10. Кетков Ю.Л. Об оптимальных методах аппроксимации плоских кривых и системе автоматизации программирования для обработки геометрической информации. // Автореферат диссертации на соискание ученой степени кандидата физико-математических наук. Горький, 1966.

Ю.Л. Кетков

Доклад был опубликован в материалах Третьей международной конференции SORUCOM-2014, 13-17 октября, Казань уже после смерти Ю.Л. Кеткова
(примеч. ред.)

Разработка диалоговых систем программирования в Нижегородском университете

В статье описываются системы коллективного пользования на базе алгоритмического языка Бейсик, разработанные в Нижегородском государственном университете им. Н.И. Лобачевского в период 1969 – 1978 гг. Первой среди них была Бейсик-система пакетной обработки, введенная в эксплуатацию на ЭВМ типа М-20 (БЭСМ-3, БЭСМ-4, М-220, М-222) в начале 1970 г. Она активно использовалась в учебном процессе и отличалась от существовавших в то время систем программирования пошаговым компилятором. Эта возможность позволила в 1971 г. преобразовать систему пакетной обработки в многотерминальную диалоговую систему коллективного пользования под названием «Студент». Система была безвозмездно передана в более чем 100 проектных организаций и учебных заведений нашей страны. В 1976 г. появилась последняя версия системы разделения времени, которая обеспечила возможность программирования в диалоговом режиме на ЕС ЭВМ под управлением ДОС ЕС.

История создания диалоговых систем программирования в Нижегородском университете началась после почти полугодовой командировки автора статьи в конце 1967 г. В это время ведущий в стране Институт Прикладной Математики АН СССР (в настоящее время ИПМ РАН им. акад. М.В. Келдыша) в соответствии с постановлением директивных органов собирал профессиональную команду программистов для создания современной операционной системы флагмана нашей вычислительной техники ЭВМ БЭСМ-6. В рамках этого постановления ИПМ планировал разработать и серию трансляторов с наиболее распространенных языков программирования. В составе сборной команды были представлены ведущие программисты наиболее крупных вычислительных центров страны и лучшие программистские кадры отдела ИПМ, возглавляемого лауреатом Государственной премии профессором М.Р. Шура-Бурой.

В рамках новой ОС БЭСМ-6 (она получила название ОС ИПМ) мне была поручена разработка одной из наиболее важных компонент системы – монитора, отвечавшего за распределение ресурсов, управление задачами и параллельными процессами, организацию очередей на выполнение внешних и внутренних запросов многочисленных исполнителей. Единственным доступным инструментом разработки был ассемблер БЕМШ, оперативно созданный под руководством В.С. Штаркмана. Однако для разработки и отладки довольно сложных и объемных программ в коде машины ничего другого, кроме запусков заданий в пакетном режиме у программистов не было. Отсутствие диалогового общения с ЭВМ сильно затягивало сроки разработки системного ПО. Поэтому нам пришлось по ходу дела познакомиться с первыми зарубежными подходами

к созданию диалоговых алгоритмических языков и соответствующих систем программирования. Конечно, это были еще не промышленные средства для профессиональных разработок, а только некоторые расширения состава операторов того или иного языка для обеспечения удобства пользователей, тяготившихся общепринятым режимом пакетной обработки заданий. На один из рабочих семинаров в ИПМ был приглашен ведущий сотрудник ВЦ АН СССР В.М. Курочкин, один из когорты первых программистов. Он познакомил нас с языком JOSS и выполненной на его базе системой разделения времени. Второй диалоговой системой оказалась реализация на ЭВМ GE-400 системы программирования на базе языка Бейсик. Источником для знакомства с ней послужило руководство пользователя, описывающего состав операторов языка программирования и нескольких директив по организации прохождения задач. Вторая система произвела на меня большее впечатление.

Возвратившись из затяжной командировки, мы (вместе со мной в Москву была направлена сотрудница отдела Н.М. Салганская, вошедшая в группу разработчиков транслятора с языка АЛМО) доложили на семинаре о результатах нашего выхода в свет. Тесные контакты с живыми кумирами программирования и предоставленная нам возможность работы на современных супер-ЭВМ (БЭСМ-6 в то время была самым мощным компьютером в Европе) произвели большое впечатление на участников семинара, которым тоже захотелось заняться чем-то более серьезным, чем поднадоевшая поддержка систем программирования на ЭВМ и разработка новых библиотечных подпрограмм. Всем очень пришлась по душе идея создания диалоговой системы программирования на базе Бейсика и возможность за счет этого обеспечить более комфортные условия в процессе обучения студентов. К постоянной работе над системой, кроме автора статьи, были подключены 4 сотрудника лаборатории автоматизации программирования – М.М. Белослудцева, Э.Н. Ильин, И.М. Куракина и В.С. Максимов. К нашей группе добровольно присоединился сотрудник НИРФИ В.В. Бочаров, который в своем институте отвечал за системное ПО (трансляторы, библиотеки стандартных программ, системный и пользовательский сервис). Первоначальное распределение обязанностей было следующим. Наиболее трудоемкую часть компилятора – блок программирования выражений, – поручили тандему Ильин-Бочаров. Дополнительно они должны были реализовать обработку операторов присваивания, определения и использования нестандартных функций. Основные процедуры ввода и вывода информации, включающие обмен с терминалом на физическом уровне и работу с блоком данных, взялся программировать В.С. Максимов. Логика обработки операторов цикла выпала на долю М.М. Белослудцевой. И.М. Куракина должна была разработать и реализовать концепцию подключения к Бейсик-программе строк, написанных на автокоде (оператор БЭСМ). На Ю.Л. Кеткова возлага-

лось общее руководство работами других членов команды, создание монитора Бейсик-системы и набора утилит, обслуживавших как системные блоки, так и рабочие программы пользователей.

В качестве устройства общения пользователя с системой в первом варианте использовалась консоль оператора – электрифицированная пишущая машинка чешского производства «Консул-254». Довольно скоро ее заменила аналогичная машинка, работавшая параллельно с консолью (чтобы не совмещать на одном устройстве рабочие места оператора ЭВМ и пользователя Бейсик-системы). К 1971 г. шумящий терминал заменили на дисплей типа «Видеотон-340», а к 1973 г. количество дисплейных терминалов в системе «Студент» было доведено до 10. В их составе были апробированы профессиональные дисплейные станции ЕС-7920. Заметный вклад в программное обеспечение, управляющее параллельным обслуживанием всех терминалов, внесли сотрудники вычислительного центра ГИФТИ Д.М. Штейман, Л.Я. Дутьшева и В.Г. Манишин.

Несмотря на то, что первая версия Бейсик-системы обеспечивала режим пакетной обработки задач, ее целью была работа в режиме реального времени, позволявшая отреагировать на любые задания 8-10 пользователей с задержкой не более 1-1.5 сек. Это означало, что время приема и обработки очередного оператора, поступившего с любого пользовательского терминала, не должно было превышать 0.15 сек. В состав этого кванта времени входили и затраты на переключение на задачу активизируемого пользователя. Точно такой же квант времени выделялся на выполнение очередного оператора в программе задачи, готовой к исполнению. За счет чего на весьма тихоходной ЭВМ типа М-20 (скорость работы соответствующих моделей варьировалась от 20 до 27 тыс. оп/с) можно было достичь такой оперативности.

Во-первых, за счет рационального использования оперативной памяти. Программе каждого пользователя выделялось «виртуальное пространство» емкостью в 4096 45-разрядных ячеек (в терминах М-20 – «куб памяти»). Полупроводниковые модели обычно комплектовались двумя кубами оперативной памяти. При этом куб с номером 1 занимала Бейсик-система, а куб с номером 0 занимала бинарная псевдопрограмма пользователя вместе со всеми данными задачи (массивами, скалярными числовыми и строковыми переменными, константами) и различными управляющими таблицами, которые были необходимы для пооператорной интерпретации псевдопрограммы. Текст исходной Бейсик-программы каждого пользователя выносился на магнитный барабан с целью экономии места в оперативной памяти задачи.

Во-вторых, нужно было свести к минимуму время переключения с одной выполняющейся задачи на другую. На ЭВМ типа М-222, полная комплектация которых могла включать 8 кубов оперативной памяти, каждая из готовых к исполнению рабочих псевдопрограмм могла не изгоняться на внешнее за-

поминающее устройство (МБ – магнитный барабан) и постоянно находиться в оперативной памяти (МОЗУ). С целью сокращения времени переключения на обслуживание очередного пользователя был определен минимальный квант работы, которую следовало выполнить для предшествующего терминала (все пользователи обладали равными правами и обслуживались по кругу).

Минимальный квант работы определялся состоянием очередной Бейсик-программы. В начальный момент времени все терминалы переводились в режим «редактирования», во время которого пользователь мог набирать очередную строку своей программы или вызвать ранее запомненный текст из архива задач. В первом случае набираемый текст накапливался в буфере терминала, и пользователю выделялся минимальный квант для синтаксического анализа введенной строки и ее преобразования в эквивалентный код псевдопрограммы только после нажатия клавиши, завершающей набор строки. Так как в Бейсик-программе почти все строки являются автономными (кроме пар FOR – NEXT и DATA – READ), то синтаксический анализ ограничивался разбором только введенной строки, и на это вполне хватало порядка 10-15 тыс. машинных тактов. Анализ парных операторов переносился на фазу выполнения псевдопрограммы. Если синтаксический анализатор обнаруживал ошибку, то на соответствующий терминал выдавалось сообщение и выполнялся переход на обслуживание очередного пользователя.

Фаза «редактирования» заканчивалась в момент приема от пользователя команды на выполнение (директива RUN), после чего минимальным квантом на следующем круге становилось время выполнения начальной или очередной строки псевдопрограммы. В отведенное время (0.15 сек.) укладывались практически все операторы Бейсика. Единственное отклонение могла вызвать работа сложной подпрограммы из библиотеки ИС-2 (например, решение системы линейных алгебраических уравнений высокого порядка). Но на такие жертвы мы шли сознательно, считая, что подобные ситуации возникают крайне редко, т.к. использование оператора БЭСМ предполагало некоторые знания в области программирования на автокоде. Да и одновременное обращение двух или более пользователей к достаточно длинным процедурам представлялось мало вероятным.

С целью минимизации времени переключения с задачи на задачу мы уделяли особое значение созданию повторно входимых (реентерабельных) программ в собственно Бейсик-системе и сокращению объема данных о задаче пользователя, загружаемых в систему перед выполнением очередного минимального кванта работы. Важную роль в соблюдении указанных критериев сыграло размещение почти всех управляющих таблиц, используемых при интерпретации псевдопрограммы, в тот же куб памяти, где в данный момент находилась и сама псевдопрограмма. При своппинге задач, ожидавших очередной обработ-

ки, их куб памяти переписывался на МБ вместе со всеми управляющими таблицами. Если две задачи одновременно находились в кубах МОЗУ, то переключение с одной задачи на другую сводилось к переключению нескольких указателей системы на управляющие таблицы очередника. В противном случае предшествующую задачу вместе с ее управляющими таблицами приходилось переписывать на МБ, а на ее место загружать новую задачу. В этом случае менять содержимое указателей на управляющие таблицы уже было не нужно, т.к. расположение таблиц во всех кубах было стандартным. Архитектура М-222, оснащенной 8-ю кубами оперативной памяти, позволяла обходиться, практически, без своппинга.

Вместо термина «исполняемая программа», которую обычно формирует компилятор алгоритмического языка, мы до сих пор использовали термин «псевдопрограмма» и ее «интерпретация». Псевдопрограмма составлялась на стадии компиляции операторов исходной Бейсик-программы и имела блочную структуру. Каждому оператору задачи соответствовал блок машинных команд в псевдопрограмме, а его интерпретация сводилась к автоматическому выполнению команд блока под управлением монитора Бейсик-системы. Таким образом, завершение работы очередного оператора исходной программы, переход к следующей строке программы или переход на строку с указанным номером контролировались монитором. Тем самым фиксировались моменты, когда исполнение текущей Бейсик-программы могло быть прервано. Важная роль в этом мониторинге принадлежала управляющей таблице операторов, содержащей 300 строк со следующей информацией:

7	14	12	12
П	N	АМОЗУ	АМЗУ

Здесь П – поле признаков

(1 в 45-м разряде: оператор исполняемый,

1 в 44-м разряде: оператор DATA,

1 в 43-м разряде: оператор END

42 - 40 разряды: признаки настройки адресов оператора БЭСМ

N – номер оператора,

АМОЗУ – адрес входа в блок псевдопрограммы,

АМЗУ – адрес начала текста исходной программы на МБ

Таблица операторов формировалась в режиме редактирования, ее строки заполнялись в порядке поступления строк исходной программы с терминала, а после завершения ввода упорядочивались по возрастанию номеров. Если в исходной программе оказалось два или более операторов с одинаковыми номерами, то строка, поступившая последней, затирала все предыдущие. Дубли таблицы операторов вместе с текстами строк исходной программы запоминались

ся на МБ. В дальнейшем это позволяло сообщить пользователю все данные об обнаруженной ошибке и произвести нужные исправления.

Ограничение на длину исходной Бейсик-программы, налагаемое таблицей операторов (не более 300 строк, т.е. не более 6 страниц машинописного текста формата А4), не слишком серьезная помеха для разработки программ на стадии первоначального обучения. Выбор цифры 300 объяснялся еще и следующей технической деталью. Для обеспечения пакетного режима первые Бейсик-программы вводились с перфокарт. А объем оперативной памяти (4096 слов) заполнялся до предела содержимым 341 перфокарты. В первой версии Бейсик-системы были предусмотрены и другие ограничения, несколько сужающие область программирования прикладных задач. Например, программа могла использовать не более 286 идентификаторов скалярных переменных, обозначаемых буквой или буквой с цифрой. Длина имени массива (одномерного или двумерного) ограничивалась одной буквой, т.е. программа могла использовать не более 26 массивов.

Конечно, с позиций сегодняшнего дня возможности того первого в стране диалогового терминал-класса могут показаться смешными. Но не надо забывать, что на наших ЭВМ в то время активно использовались трансляторы только с полного (ТА-2) или урезанного Алгола (ТА-1), еще только начинал внедряться транслятор с Фортрана (Ф-20). И все они эксплуатировались в пакетном режиме, т.е. на обнаружение одной-двух первых ошибок на стадии отладки уходили сутки. Да и время прохождения задачи через ту или иную систему программирования исчислялось десятками минут. А в диалоговой системе программирования за один проход можно было выловить почти все синтаксические ошибки. И вместо месяца пребывания в бесконечных пакетных очередях справиться с простым студенческим заданием за два-три сеанса работы за терминалом. Статистика за два семестра одного учебного года (1975/76) показала следующее. В первом семестре терминал-класс обеспечивал занятия 24 академических групп (~ 550 первокурсников). За 1300 сеансов было отлажено порядка 1200 задач. Во втором семестре через терминал-класс прошли уже 30 академических групп (~ 700 студентов). За 1000 сеансов было отлажено порядка 1500 задач. Процессорное время, затраченное на набор и отладку программ, составило всего 40 часов машинного времени М-222. Эти цифры не идут ни в какое сравнение с производительностью пакетного режима обработки заданий.

Наши системы не были в прямом смысле интерпретаторами языка Бейсик, как первые, да и не только первые зарубежные Бейсик-системы. Visual Basic, появившийся на IBM совместимых ПК в 1991 г. до версии 4.0 был полным интерпретатором, замедлявшим работу исполняемых программ в 20-30 раз.

В разработанных версиях Бейсик-систем с самого начала была включена возможность программирования в терминах, близких к автокоду ЭВМ типа

М-20. Это обеспечивало использование любой стандартной программы из библиотеки ИС-2 (ИС-22). Общий объем диалоговой системы не превышал 12000 45-разрядных слов (т.е. порядка 60 Кб в современном исчислении). На ее создание мы потратили порядка 5-6 человеко-лет. При тиражировании последующих версий даже на ЭВМ с другой архитектурой (например, ЕС ЭВМ) затраты снизились почти вдвое. Сказались приобретенный опыт, отработанные схемы анализа и реализации операторов Бейсика и, главным образом, накопленная библиотека соответствующих макроопределений на языке ассемблера. Переписать ее из кода М-20 в код ЕС ЭВМ оказалось не так уж и сложно.

На ежегодных конференциях пользователей ЭВМ типа М-20 (комиссия КЭВМ возглавлялась М.Р. Шурой-Бурой) сообщения об очередной версии Бейсик-системы вызывали неизменный интерес и приводили к регулярному появлению новых пользователей. В те времена распространение оригинального ПО производилось на безвозмездной основе. В списке заявок на получение и установку диалоговых систем программирования, разработанных в НИИ ПМК, фигурирует более 100 проектных организаций и вузов нашей страны. В Нижегородском университете Бейсик-система функционировала около 10 лет до тех пор, пока не началось увлечение персональными компьютерами.

Первое самиздатовское руководство по Бейсик-системе появилось в НИИ ПМК в 1971 г. В ближайшие пару лет были представлены ряд сообщений о системе на Всесоюзных конференциях и Международном конгрессе болгарских математиков. Первая публикация о Бейсике на русском языке в центральных издательствах состоялась с двухлетним опозданием в 1978 г. Автору статьи принадлежит, наверное, самая большая коллекция книг по Бейсику в нашей стране.

Литература

1. Белослудцева М.М., Бочаров В.В., Ильин Э.Н. и др. BASIC-система пакетной обработки задач на ЭВМ типа БЭСМ-4, М-220 (инструкция по эксплуатации), – Горький, НИИ ПМК, 1971.
2. Белослудцева М.М., Бочаров В.В., Ильин Э.Н. и др. Использование алгоритмического разговорного языка BASIC на ЭВМ типа БЭСМ-4. // Труды II Всесоюзной конференции по применению ЭВМ и математических методов в планировании и управлении предприятием. – Горький, октябрь, 1971.
3. Белослудцева М.М., Бочаров В.В., Ильин Э.Н. и др. Использование алгоритмического разговорного языка BASIC на ЭВМ типа БЭСМ-4, М-220Б М-222. // Труды симпозиума «Теория языков и методы построения систем программирования». – Киев-Алушта, 1972

-
4. Кетков Ю.Л. BASIC-система пакетной обработки задач на ЭВМ типа БЭСМ-4, М-220. // Трети конгрес на българските математици. – Варна, 6-15 септември, 1972
 5. Белослудцева М.М., Бочаров В.В., Ильин Э.Н. и др. BASIC-система пакетной обработки задач на ЭВМ типа БЭСМ-4М, М-220, М-222. Учебное пособие (под ред. Ю.Л. Кеткова). – Горький, 1973. – 193 с.
 6. Дутьшева Л.Я., Манишин В.Г., Штейман Д.М. Совместимая система разделения времени «Студент». // Управляющие системы и машины, 1975, №1
 7. Дутьшева Л.Я., Кетков Ю.Л., Кузин С.Г. Обучение программированию в системе коллективного пользования «Студент». // В сб. Машинное обучение с помощью диалога. – М., материалы семинара общества «Знание», 1976. – с. 95-98
 8. Кетков Ю.Л., Рябов А.Н. Программирование на БЭЙСИКЕ (Входной язык системы программирования Бейсик-222). Учебное пособие. – Горький, 1977. – 73 с.
 9. Кетков Ю.Л. Программирование на Бэйсике. – М.: Статистика, 1978. – 158 с.
 10. Кетков Ю.Л., Максимов В.С., Рябов А.Н. Введение в системное программирование на языке Ассемблера ЕС ЭВМ. – М.: «Наука», Гл. ред. физ.-мат- лит., 1982. – 264 с.
 11. Кетков Ю.Л., Куракина И.М. Программирование на алгоритмических языках Бейсик и Фортран. – Горький: изд. ГГУ, 1983. – 104 с.
 12. Зверев В.И., Кетков Ю.Л., Максимов В.С. Алфавитно-цифровые дисплеи ЕС-7920 в диалоговых системах. – М.: «Наука», Гл. ред. физ.-мат. лит., 1986. – 238 с.
 13. Кетков Ю.Л., Куракина И.М. Программирование на алгоритмических языках Бейсик и Фортран. – Горький: изд. ГГУ, 1986. – 135 с.
 14. Кетков Ю.Л. Диалог на языке Бейсик для мини- и микро-ЭВМ. – М.: «Наука», Гл. ред. физ.-мат. лит. 1988. - 368 с.
 15. Кетков Ю.Л. Толковый словарь языка программирования Бейсик. – М.: «Наука», Гл. ред. физ.-мат. лит. 1992. – 320 с.
 16. Кетков Ю.Л. GW-, Turbo- и Quick-BASIC на IBM PC/XT и AT. – М.: Финансы и статистика, 1992. – 240 с.
 17. Кетков Ю.Л., Кетков А.Ю., Шапошников Д.Е. Школьная энциклопедия: Персональный компьютер. – М.: Большая Российская Энциклопедия, изд. Дом Дрофа, Москва, 1997, 1998. – 440 с.

18. Кетков Ю.Л., Кетков А.Ю. Практика программирования: Бейсик, Си, Паскаль. Самоучитель. – Санкт-Петербург: БХВ-Петербург, 2001, 2002. – 480 с.
19. Кетков Ю.Л., Кетков А.Ю. Практика программирования: Visual Basic, C++ Builder, Delphi. Самоучитель. – Санкт-Петербург: БХВ-Петербург, 2002. – 464 с.

Ю.Л. Кетков

Доклад был опубликован в материалах Третьей международной конференции SORUCOM-2014, 13-17 октября, Казань уже после смерти Ю.Л. Кеткова
(примеч. ред.)

Информатизация в средней школе на базе алгоритмического языка БЕЙСИК

БЕЙСИК – алгоритмический диалоговый язык, ориентированный на широкий круг пользователей. Аббревиатура BASIC происходит от начальных букв английских слов, переводимых как «Многоцелевой язык символьных инструкций для начинающих».

Разработан он в 1964 г., на отечественных ЭВМ культивируется с начала 70-х годов. С появлением вычислительной техники третьего поколения - ЕС и СМ ЭВМ, микро-ЭВМ и микропроцессоров БЕЙСИК превратился в один из наиболее популярных языков программирования. Был период, когда на микро-ЭВМ кроме ассемблера и БЕЙСИКА ничего другого не было.

Он широко используется в системах автоматизации научных исследований, в инженерно-технических расчетах, в медицинской диагностике и т.п.

В ряде стран ближнего и дальнего зарубежья изучение БЕЙСИКА составляет основу школьных курсов программирования для старших классов. До недавнего прошлого БЕЙСИК широко использовался и в вузах для обучения азам компьютерной грамотности.

Как человек, причастный к становлению, развитию и пропаганде БЕЙСИКА в нашей стране, я хотел бы остановиться трех следующих моментах:

- как отразилось на БЕЙСИКЕ бурное развитие средств вычислительной техники за последние годы ?
- в чем причины такой популярности этого языка и почему он продолжает оставаться базовым для многочисленных курсов по информатике ?
- какую роль сыграл Нижегородский университет в становлении этого языка в нашей стране ?

Пожалуй, ни один алгоритмический язык не вызвал такого резкого разделения в обществе на ярых его приверженцев и столь же непримиримых его противников. По всей видимости, начало нелицеприятных высказываний о БЕЙСИКЕ в открытой печати было положено известным голландским специалистом Э.В. Дейкстрой, который в приложении к журналу АСМ объявил умственно оболваненными и лишенными надежды на исцеление всех студентов, чье знакомство с программированием начиналось с БЕЙСИКА.

Публикацию подобного рода заявлений охотно подхватил советский журнал «Информатика и образование», который в одном из своих номеров за 1986 г. привел подборку наиболее негативных цитат (А. Дуванов). Вслед за этим последовало высказывание Я. Зайдельмана, сделанное в духе массовых кампаний периода перестройки : «БЕЙСИК действует на мозг подобно алкоголю или наркотикам, нанося непоправимый вред даже в малых дозах» (ИО, 5, 1987).

Стиль наших соотечественников и выбираемые ими выражения носят не очень парламентский характер. Но самое печальное в этих высказываниях – отсутствие какого-либо анализа языка и тенденций его развития. Как правило, критике подвергается отдельная компонента языка, конкретная версия или не очень удачная реализация соответствующей системы программирования. И все это автоматически переносится на БЕЙСИК в целом. А ведь этот язык гораздо мобильнее своих маститых коллег, он активно развивается, и слабых мест у него становится все меньше и меньше.

Общественная вражда сторонников и противников БЕЙСИКА больше напоминает известное противостояние тупоконечников и остроконечников. Мне было очень интересно наблюдать изменение отношения к этому языку академика Андрея Петровича Ершова, внесшего огромный вклад в информатизацию нашего общества. Как профессиональный программист, воспитанный в духе строгих алголоподобных концепций, он достаточно пренебрежительно отозвался о БЕЙСИКЕ на одной из первых конференций по программированию. Однако в последние годы своей жизни он организовал цикл телевизионных лекций по информатике и использовал именно БЕЙСИК для иллюстрации приемов программирования.

В завершение хотелось бы привести достаточно забавное высказывание, появившееся в сборнике «Экспертные системы» (1987 г.). Речь в нем идет об освоении алгоритмического языка ПРОЛОГ, используемого для разработки систем искусственного интеллекта. «Если вам непременно необходимо использовать ПРОЛОГ, то подождите пока он не будет «проглочен» языком БЕЙСИК, этим питоном, пожирающим все, что лежит на его пути. БЕЙСИК только что закончил «переваривать» язык ПАСКАЛЬ со всеми его управляющими структурами. После небольшой паузы и нескольких «отрыжек» он будет в состоянии «слопать» ПРОЛОГ, так что мы увидим варианты языка БЕЙСИК со встроенным механизмом доказательства теорем методом резолюций».

Эволюция языка БЕЙСИК с удивительной последовательностью повторяет этапы развития средств вычислительной техники.

Первое поколение БЕЙСИК-систем связано с открытием эры диалога с ЭВМ. Вообще говоря, первая диалоговая система программирования, разработанная в 1963 г., использовала в качестве входного языка JOSS. Однако дальнейшего развития этот язык не получил, поэтому становление диалогового программирования связывают с БЕЙСИКОМ. Его первая редакция и соответствующий интерпретатор появились в 1964 г. благодаря энтузиазму группы студентов Дартмутского колледжа, возглавляемой профессорами Джоном Кемени и Томасом Куртцем. Этому коллективу предстояло решить две основные задачи:

-
- ввести в язык и систему программирования процедуры диалогового взаимодействия с пользователем;
 - создать максимально простой язык, который бы легко осваивался пользователями и одновременно допускал бы простую программную реализацию.

Известный персонаж книги «Двенадцать стульев» Элочка Щукина легко и свободно обходилась 30 словами, фразами и междометиями, придирчиво выбранными ею из всего великого, многословного и могучего русского языка. Вспомните знаменитые изречения: «Не учите меня жить», «Хамите», «У вас вся спина белая».

Первая версия БЕЙСИКА насчитывала всего порядка 25 слов, немного опережая по многословию академический АЛГОЛ-60.

Однако в идеологическом плане новый язык оказался существенно проще. Структура его операторов была максимально приближена к формату машинных команд: номер строки - адрес команды, служебное слово - код операции, тело оператора - аналог адресной части. Это позволило построить сравнительно простой интерпретатор, повторяющий логику работы ЭВМ и допускающий более гибкое построение программы - вставку строк с промежуточными номерами, замену или удаление ошибочных строк, обход невыполняемых процедур и т.п. Непосредственный режим выполнения операторов, набранных без номера строки, превращал БЕЙСИК-интерпретатор в мощный калькулятор.

Шесть Дартмутских редакций (1964-1971 гг.) определили основные направления развития языка. Минимальное ядро начало обрастать такими расширениями как набор матричных операций, являющихся одним из важных инструментов радиофизика и электротехника, процедурами обработки текстовой информации обмена с файлами данных, без чего не обходится ни одна из сфер управления.

В конце 60-х начале 70-х гг. БЕЙСИК получил мощную поддержку со стороны фирм General Electric, Hewlett-Packard и Digital Equipment Corp. Для двух последних этот период совпал с крупными успехами на рынке мини-ЭВМ.

Бум персональных ЭВМ, обусловленный массовым появлением дешевых 8- и 16-разрядных микропроцессоров вдохнул вторую жизнь и в БЕЙСИК. К появлению систем второго поколения больше других причастна фирма Microsoft Corp. Ее основатель и бессменный президент Билл Гейтс, являющийся сейчас одним из самых молодых миллиардеров США, в середине 80-х годов разработал интерпретатор BASIC-80, ориентированный на эксплуатацию в широкой гамме бытовых и учебных ПЭВМ, построенных на базе наиболее популярных микропроцессоров Zilog-80 и Intel-8080. С появлением новых периферийных устройств – графических дисплеев, программируемых звуковых генераторов, таймеров, игровых пультов, – эта версия расширялась. На ПЭВМ

тех лет эксплуатируется довольно много модификаций в той или иной мере повторяющих ставший de-facto стандартом для бытовых и учебных компьютеров MSX-BASIC.

Успехи БЕЙСИКА на рынке мини- и микро-ЭВМ заставили ряд ведущих фирм пересмотреть свое отношение к этому языку и организовать серьезные исследования по развитию его функциональных возможностей. В первую очередь эти перемены коснулись такого гиганта вычислительной индустрии как IBM. Слегка подразнив своих пользователей не очень зрелым БЕЙСИКОМ на одной из ранних систем разделения времени TSO, сотрудники фирмы IBM реализовали в рамках операционной системы виртуальных машин довольно мощный проект BASIC-IBM/VM. Последний имел словарный запас, почти в 10 раз превышающий одну из ранних Дартмутских версий. Один только справочник по системе насчитывал более 400 экранных страниц.

С появлением серии профессиональных 16-разрядных ПЭВМ БЕЙСИК-интерпретатор фирмы Microsoft встраивался в материнскую плату в виде прошитой микросхемы ПЗУ и составлял неотъемлемую часть системного программного обеспечения. В результате конкуренции между двумя основными производителями программных продуктов - фирмами Microsoft и Borland Int. – появились две интегрированные среды программирования Quick- и Turbo-BASIC.

Их почти пятилетняя борьба, завершившаяся выбыванием Borland Int., привела к появлению БЕЙСИК-систем третьего поколения. Уровень комфорта, предоставляемый пользователю такими средами, практически, не отличается от возможностей профессиональных языков типа Си или Паскаль. Интересно отметить, что к третьему поколению примыкает и версия «Истинного БЕЙСИКА» (True BASIC), к разработке которой вернулись почти через 25 лет его создатели - J. Kemeny и T. Kurtz.

Как революцию в программировании под Windows расценивает журнал PC Magazine появление очередного детища фирм Microsoft – системы Visual BASIC for Windows. Это название можно было бы перевести как «Наглядный БЕЙСИК». По итогам 1991 г., подведенным наиболее популярными в мире журналами по программному обеспечению, Visual BASIC вошел в пятерку лучших программных продуктов. В 1992 г. фирма Microsoft выпустила уже вторую версию VB/Win, которая поставляется Стандартным Изданием по цене 195\$ для пользователей, изредка программирующих свои задачи, и Профессиональным Изданием по цене 495\$ для профессионалов.

Наглядный БЕЙСИК можно смело отнести к продукту четвертого поколения. С его помощью создаются прикладные программы, включающие стандартный интерфейс с пользователем – как раз то, чего не хватает многим на-

шим программным продуктам. На Западе-то, как и у нас, тоже встречают по одежке, и без хорошего оформления на рынок программ нечего соваться.

Наглядный БЕЙСИК снабжен самыми современными средствами отладки с очень удобным управлением. Он поддерживает работу с коммуникационными портами и снабжен средствами для управления электронной почтой.

Подводя итоги краткому историческому экскурсу, хочу отметить наиболее важные положительные моменты, определившие ведущие позиции БЕЙСИКА в информатизации средней, а до недавних пор и высшей школы:

- во-первых, простота освоения. Начальные навыки работы, практически, с любой БЕЙСИК-системой приобретаются после двух-трех занятий;
- во-вторых, БЕЙСИК в том или ином объеме реализован на каждой ЭВМ, доступной средним и высшим учебным заведениям;
- в-третьих, БЕЙСИК располагает средствами для обработки текстовой и графической информации, позволяет включать в программу элементы компьютерной музыки, предоставляет возможность создавать игровые программы;
- наконец, немаловажным обстоятельством является наличие достаточно большого количества разнообразной и вполне доступной литературы по, практически, всем версиям БЕЙСИКА.

И, в заключение, несколько слов о роли Нижегородского университета в становлении БЕЙСИКА в нашей стране. Многие из работ в области развития вычислительной техники и программного обеспечения, выполненные в стенах нашего университета, можно сопроводить термином «пионерские».

Здесь была разработана и введена в эксплуатацию цифровая вычислительная машина ГИФТИ – первая ЭВМ в нашем городе. На ее базе в 1957 г. был создан первый вузовский вычислительный центр. В ГГУ был открыт первый в стране факультет вычислительной математики и кибернетики.

В 1969-70 гг. здесь же была разработана первая в стране БЕЙСИК-система для отечественных вычислительных машин типа М-20. В отличие от существовавших в то время зарубежных систем она использовала принцип компиляции (прошу не путать этот термин с заимствованием), а не интерпретации, что позволило на 1-2 порядка повысить общую производительность ЭВМ.

На ее базе в 1973 г. был введен в строй первый в стране учебный терминал-класс с 11-ю терминалами, прослуживший верой и правдой около 10 лет.

Первая и далеко не единственная книга по БЕЙСИКУ, опубликованная в нашей стране принадлежит автору настоящего доклада (1978 г.). Последняя из них вышла в 2002 г. и посвящена одной из наиболее популярных версий, эксплуатирующихся на IBM PC, – Visual BASIC.

Ю.Л. Кетков

Доклад прочитан и опубликован не был (*примеч. ред.*)

Запросы на установку транслятора Бейсик

Всего подобных заявок было более 150

ЎзССР ФАНЛАР АКАДЕМИЯСИ

Ўзбекистон „Кибернетика“
илмий-ишлабчиқариш бирлашмаси

Меҳнат Қизил Байроқ ордени
ҲИСОБЛАШ МАРКАЗЛИ
КИБЕРНЕТИКА ИНСТИТУТИ
700125, Ташкент, Ф. Ходжаева кўч., 34 уй



АКАДЕМИЯ НАУК ЎзССР

Ўзбекское научно-производственное
объединение „Кибернетика“

Ордена Трудового Красного Знамени
ИНСТИТУТ КИБЕРНЕТИКИ
С ВЫЧИСЛИТЕЛЬНЫМ ЦЕНТРОМ
ул. Ф. Ходжаева, 34; т. 625737, 627247, 627043

18.06.81
№ 6062/1428-20

Директору НИИ прикладной
математики и кибернетики
при ГГУ проф.Сергиевскому А.В.

Уважаемый Андрей Владимирович!

В Институте кибернетики с ВЦ НПО "Кибернетика" АН УЗССР ведется цикл исследования по математическому моделированию функциональных подсистем организма. Технической базой реализации математических моделей является : ЭВМ "М-222", установленная в клинике.

В целях более полного использования возможностей данного типа ЭВМ, нам хотелось бы дополнить имеющиеся трансляторы языков АЛГОЛ и Фортран для данной машины, транслятором с языка BASIC . Известно, что в Вашем институте, под руководством к.ф-м.н. Елия Лазаревича Кеткова был разработан такой транслятор в семидесятых годах.

Если сейчас есть возможность передачи копии транслятора языка BASIC для использования в УзНПО "Кибернетика", просьба сообщить условия выполнения нашей просьбы.

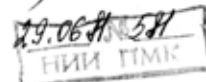
Зам.генерального
директора УзНПО "Кибернетика"
по науке, д.ф-м.н., проф.

T. Kerpuly
2/11/81
Бондаренко

/Бондаренко Б.А./

Зав.лабораторией
"Медицинская кибернетика", к.т.н.

Ф.Т.Аддылова
Аддылова Ф.Т./



Тип. «Фан» УЗССР — 799—5000—78 г.

124

ПРИЛУЧЕНИЕ
№ 12
6/1-72.



ВОЙСКОВАЯ ЧАСТЬ
09703

19 декабря 1971
№ 2743

ДИРЕКТОРУ НАУЧНО-ИССЛЕДОВАТЕЛЬСКОГО
ИНСТИТУТА ПРИКЛАДНОЙ МАТЕМАТИКИ И
КИБЕРНЕТИКИ.

Г. Горький, Ульянова 10.

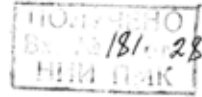
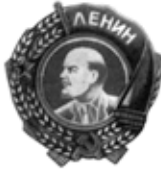
Войсковая часть 09703 заинтересована Вашей работой по
реализации языка BASIC , о котором был сделан
доклад на последней конференции по математическому
обеспечению ЭВМ типа "М-20".

Прошу выслать описание языка BASIC , инструк-
цию по использованию транслятора и комплект перфокарт.

ЗАМ. КОМАНДИРА В/Ч 09703

Shurmel'ev
/ ШУРМЕЛЕВ /

Северодвинск , в/ч 09703



ОРДЕНА ЛЕНИНА
ПРЕДПРИЯТИЕ
п-я А-3556

МК-07-54/

Экз. № 1

394013, г.Воронеж

21 января 1974 г.

Иск Кетков
28.01.74

г.ГОРЬКИЙ
Горьковский ун-т им.Лобачевского
Научно-исследовательский ин-т
прикладной математики : информатики
и кибернетики.

Прошу выслать 10 экз-ров учебного пособия "Basic"- система пакетной обработки задач на ЭВМ типа БЭСМ,4, М-220, М-222" под ред. канд. физ.мат.наук Кеткова И.Л.

Оплата гарантируется.Зенез направить по адресу: 394013,г.Воронеж, предприятие п/я А-3556, технической библиотеке.

ЗАМ.РУКОВОДИТЕЛЯ ПРЕДПРИЯТИЯ (ГОЛУБЕВ А.А.)

ЗАМ ГЛ.БУХГАЛТЕР

(ЗОЛОТАРЕВ Н.И.)



ПОЛУЧЕНО
Вх. № 107 от 24.12.72
НИИ ПМК СССР

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ
МОСКОВСКИЙ ОРДЕНА ЛЕНИНА АВИАЦИОННЫЙ ИНСТИТУТ
ИМЕНИ СЕРГО ОРДЖОНИКИДЗЕ

Москва А-80, А/л 315

Волоколамское шоссе, 4

158-00-02

Ваш индекс

Дата

Наш индекс ВП/16 6 декабря 1972 г.

РЕКТОРУ ГОРЬКОВСКОГО
ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

Ю. А. Керснов
28.12.72

Просим Вас учесть в плане выпуска I-ой и 2-ой части Инструкции «*Васе*» - система пакетной обработки программ для машин М-220, БЭСМ-4 нашу заявку на 10 экз. каждой из частей инструкции и выслать указанное количество экз. по адресу: 125871 ГСП Москва А-80, Волоколамское шоссе, 4. Московский авиационный институт, Вычислительный центр.

Оплату гарантируем. Наш расчетный счет 140212 в Ленинградском отделении Госбанка г.Москвы.

ПРОРЕКТОР ИНСТИТУТА *Сонин* (А. ЛЕБЕДЕВ)

И.О.ГЛ.БУХГАЛТЕРА

Шулятикова (ШУЛЯТИКОВА К.И.)

С. К. Мобильный

И

20 12 72

ФОТОГРАФИИ



Школа №8, 5 класс «А», г. Горький, 1947 г.



Зелёный город, пионерлагерь горкома ВЛКСМ, лето 1949 г.



Август 1950 г., село Гоневка, 9-й класс



Одноклассники. 10-й класс «Б», школа №8, июнь 1952 г.



Картошка, г. Клин, 1953 г.



Бурлаки на Волге. Горький - Казань, июль 1956 г.



О.С. Фрум с детьми Ю.Л. Кетковым и Р.Л. Фрум-Кетковым



Р.Л. Фрум-Кетков, Ю.Л. Кетков, Я.И. Коренман, 1-й курс, 1953 г.



Братья Л.И Кетков и И.Д. Червенко



Отец Ю.Л. Кеткова - Лазарь Иванович



И.Д. Червенко с внуками и родственниками, в том числе детьми Ю.Л. Кеткова - Сергеем и Александром



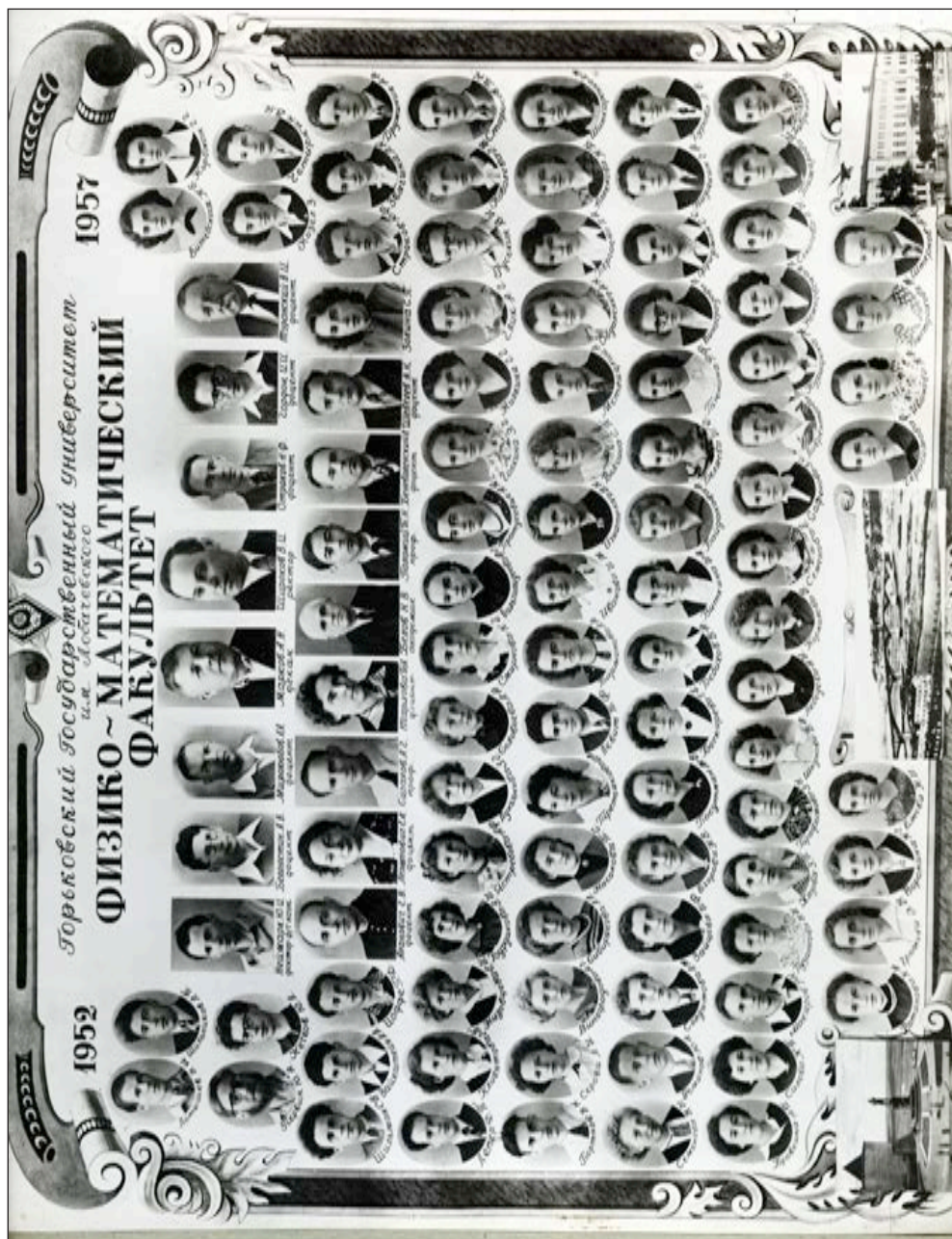
После сдачи Неймарка, июнь 1956 г.



Стадион «Динамо», 1956 г.



Ю.Л. Кетков - чемпион города Горького по прыжкам в высоту, май 1956 г.



Выпуск физмата ГГУ 1957 г.



Ю.Л. Кетков и Ю.А. Первин



Артиллеристы на привале



Общежитие МГУ. 1957 г.



Целина



Зимой 1965 г.



Вторая всесоюзная конференция пользователей ЕС ЭВМ, Москва, 1978 г.



Ввод данных с перфокарт, БЭСМ-3М



С.И. Альбер, к.ф.-м.н., зав. отделом НИИ ПМК



За работой на БЭСМ-3М



В НИИ ПМК



Ю.Л. Кетков и «железный Феликс» (механический арифмометр Феликс). Фото с юмором - на Доску почёта



На математическом конгрессе в Варне, 1967 г.



Занятия в научной школе на берегу Байкала



Всесоюзный симпозиум по методам реализации алгоритмических языков



4-й отдел НИИ ПМК, 1972 г.



4-й отдел НИИ ПМК, 1984 г.



Перед погружением



Крымский охотничий трофей



Предмет вожделения



Семья. Слева направо: Ида Владимировна, Александр, Сергей



С сыном Сергеем



С сыном Александром



Поход с семьёй за грибами. Слева направо: Сергей, Ида Владимировна, Александр



Добыча



За грибами со старшим внуком Сергеем



Любимые внуки - Катя и Юля



Первый раз в первый класс идёт внук Сергей. Слева направо: И.В. Кеткова, С.С. Кетков, С.Ю. Кетков, Л.А. Кеткова. Фото Ю.Л. Кеткова



Любимый кот Касьян. Фото Ю.Л. Кеткова



Пельмени - любимое блюдо. Мастер на все руки



В домашней обстановке



Друзья и коллеги: Ю.Л. Кетков и Г.Г. Денисов



Коллеги. Слева направо: Ю.Л. Кетков, В.Н. Фокина, В.Н. Шевченко



Преемственность поколений. Слева направо: В.П. Гергель, Р.Г. Стронгин, Ю.И. Неймарк. Фото Ю.Л. Кеткова



В роли тамады



В НИИ ПМК. Слева направо: Ю.Г. Васин, Ю.Л. Кетков, Г.Г. Денисов



Государственный экзамен



Со студентами ВМК



Награждение победителей. М.М. Шульц и Ю.Л. Кетков



Картографические приложения



Кафедра МО ЭВМ. На первом плане слева направо: Ю.Л. Кетков, Р.Г. Стронгин, В.П. Гергель



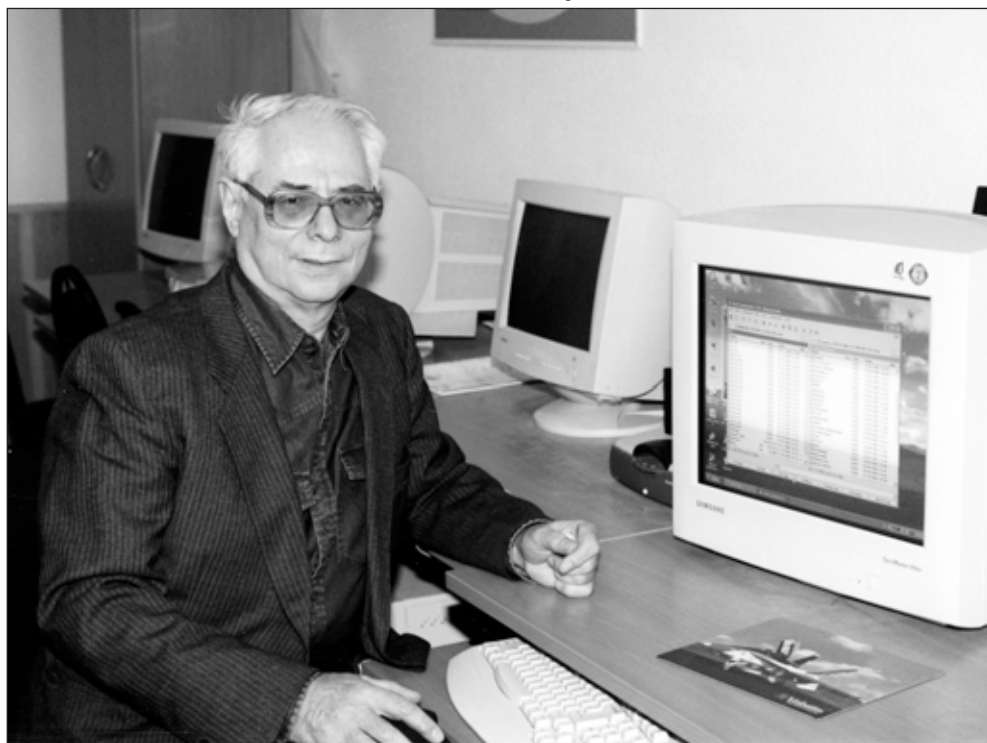
На совещании. Слева направо: Ю.Л. Кетков, В.П. Савельев, Р.Г. Стронгин



4-й отдел НИИ ПМК



На работе



В терминал-классе

**ВОСПОМИНАНИЯ
О Ю.Л. КЕТКОВЕ**

Близкое и далекое

Суммируя деловые качества Юлия Лазаревича, и, зная его долгие годы в период руководства НИИ ПМК, профессор Ю.Г. Васин употребил ключевое слово «первопроходец». Действительно, вся научная карьера Ю.Л. складывалась так, что ему приходилось многое делать впервые, об этом убедительно свидетельствует список его публикаций. Но будет справедливо сказать, как это верно и по отношению к его индивидуальным особенностям, когда чисто пионерский характер, в подлинном смысле этого слова, определял его поступки на протяжении всей жизни.



Мы познакомились весной 1957 г. среди шумной тесноты маленькой комнаты комитета комсомола Горьковского университета на ул. Свердлова 37. Страна одержимо готовилась к международному фестивалю молодежи и студентов в Москве, повсюду проходили мини фестивали, каждый факультет находился в творческом поиске. Мне как секретарю факультетского бюро ВЛКСМ истфила по статусу полагалось быть ближе к центру событий. Атмосфера азарта в предстоящей борьбе за первое место, столкновение амбиций, соревнование талантов - все создавало приподнятое настроение. Даже на этом фоне его нельзя было не заметить.

Юля был «москвич» конца 50-х. Свободен и независим в своих суждениях, он располагал к себе не только стилем поведения, но внутренним превосходством. Очень скоро тайное стало явным: он работал над своим дипломом в «святая святых» новой научной дисциплины, в лаборатории управляющих ЭВМ в отделении прикладной математики МИ АН СССР. Его научным руководителем был сам А.А. Ляпунов, хотя практически, как я узнала позднее, его работу контролировал Э.З. Любимский, который только что защитил первую в стране кандидатскую диссертацию по вычислительной математике. Это было событием в математическом мире, ибо данная отрасль знаний наконец-то получила признание среди математиков-«фундаменталистов». Вычислители праздновали победу. Отсвет этих событий явно отражался в его настроении. Теперь предстояло закончить дипломную работу и завершить годичную стажировку в МГУ вместе с другими пятью студентами физмата. Защита диплома, которая проходила в закрытом режиме, и тесный контакт с «корифеями программирования» как называл Ю.Л. сотрудников лаборатории, не только определили его дальнейший жизненный путь, но и оставили добрый след в памяти его кураторов.



Во время работы над ОС ИПМ

Во всяком случае, когда в 1967 г. по решению Президиума АН СССР была создана комиссия по разработке современного программного обеспечения БЭСМ-6 и приглашались квалифицированные специалисты, Ю.Л. был в их числе. Группой руководил М.Р. Шура-Бура, а в исполнительный состав входили Э.З. Любимский, И.Б. Задыхайло, С.С. Камынин, Ю.Л. Кетков, Д.А. Корягин, В.С. Штаркман. Естественно, я не могу судить ни о сути, ни о деталях их общей работы, но опосредованно, через его письма и рассказы я пыталась составить представление о коллективном портрете этих незаурядных и ярких личностей. Типичные представители шестидесятников, они были сравнитель-

но молоды, талантливы, способны на научное подвижничество и в то же время профессионально готовы к решению самых сложных проблем, связанных с космосом. У них было много общего.

К сожалению, почти все они ушли достаточно рано, в 90-е и в начале 2000-х гг. Ю.Л. успел отдать дань памяти этим замечательным людям в своем докладе на международной конференции, посвященной истории вычислительной техники и программного обеспечения в СССР, состоявшейся в Великом Новгороде в 2011 г. Можно было видеть, как тщательно и ответственно Ю.Л. собирал материал для этого доклада. Он заново переживал эпизоды совместной работы, радовался находкам, вспоминал обстановку, настрой на рождение и обмен новыми идеями, когда вырабатывался и соответствующий стиль общения. Их команда в ОПМ была заметной не только по результатам. Розыгрыш и юмор входил в работу над программами как составная часть. Однажды мне по почте пришли усы, так я поняла, что работа по введению в эксплуатацию системы была завершена, ибо вся группа дала обет не бриться, пока не финишируют с победными результатами.

Однако до упомянутых событий, связанных с работой в Москве, расположился существенный этап длиной почти в десять лет. Распрощавшись с МГУ, Ю.Л. готовился вместе с остальными ребятами своего курса к военным лагерям. Ни адреса лагеря, ни время прибытия будущим офицерам не сообщили. В моем плане стояла преддипломная практика в фондах столичных библиотек, а затем поездка на целину. Надвигающаяся разлука воспринималась как катастрофа. Тогда еще не пели «с любимыми не расставайтесь», но ощущения были те же. После возвращения из Москвы в июне я решила объехать все расположенные в области точки дислокации военных лагерей, пока не подсказали заглянуть в Петушки под Владимиром. Нашла и простилась. Но к отправлению эшелона на целину он успел вовремя. И все равно настроение было грустное. Из всех дверей «телячьих вагонов» доносилось: «... сиреневый туман над нами проплывает... кондуктор не спешит, кондуктор понимает, что с девушкой я прощаюсь навсегда...».



По дороге на целину

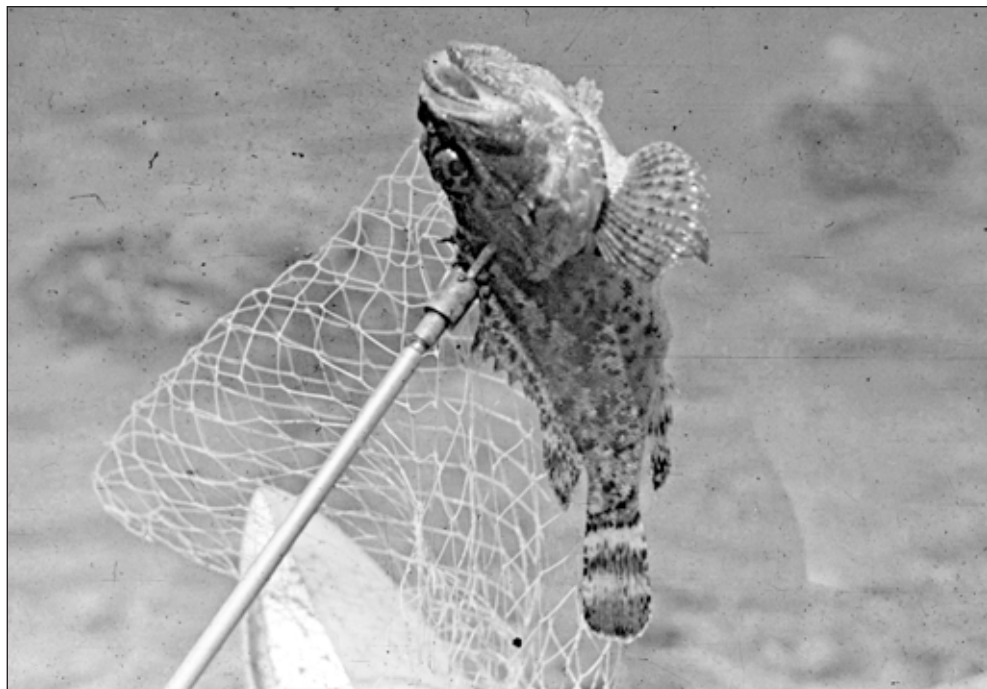
Наши целинные будни в Алтайском крае прекрасно описаны многими сокурсниками. 8 бригада истфила расположилась в полевом стане в 20 км от деревни Волчна Бурла. Связь с миром - только посредством радио и самосвала. Можно представить, какой праздник подарил нам Ю.Л., прислав по почте ящик дорогих конфет. Он всегда чутко воспринимал проблемы других, попавших в трудные условия, как собственные. Уже потом выяснилось, что деньги на такой недешевый подарок он заработал на стройке. Еще большую радость принесло сообщение, что он вместе с группой молодых сотрудников университета тоже едет на Алтай. Радость оказалась преждевременной, ибо расстояние в 300 км, которое отделяло их бригаду от нашей, оказалось непреодолимым. О его трудовых подвигах я узнала от его товарищей, ибо сам он никогда не любил говорить о собственной персоне. Вот несколько эпизодов. Однажды ночью, когда после разгрузки вагонов с цементом уставшие и грязные, ребята обнаружили, что шофер пьян, Ю.Л. молча сел за руль самосвала, хотя делал это впервые. Степь, гладкая как стол, позволяла снизить риски до минимума. Довез без потерь. А буквально на следующий день совхозный бригадир предложил ему ввести самосвал с зерном на весы. Тут нужна была почти ювелирная точность, чтобы не сбить ворота. Ему всегда нравилось овладевать незнакомым делом,

он как бы выигрывал спор с самим собой. В этом я убедилась на следующий год на берегу Черного моря. К тому времени я уже имела представление о его характере и с опасением прикидывала, как пройдет наш отдых. Лежание на пляже в виде подкопченной тушки исключалось полностью. Что меня ждет? И он нашел дело. В конце 50-х годов подводное плавание в Советском Союзе еще только начинало осваиваться отдельными энтузиастами, но была заложена хорошая теоретическая база в виде переведенной и отечественной литературы, например книга О. Хлудовой, экземпляр которой с засохшими каплями морской воды до сих пор хранится на книжной полке как дорогая реликвия. Страсть к подводному плаванию захватила Ю.Л. сразу и надолго. Тут же приобрели первые отечественные не очень удобные маски и трубки, неподъемные ласты. Все события происходили в Лазаревском, недалеко от Сочи. Тогда это место не было фешенебельным курортом, а располагало чуть ли не единственной пыльной улицей, ведущей к морю. Вскоре выяснилось, что Ю.Л. без очков почти ничего не видит даже в прозрачной воде. А цель погружений была определена заранее – собирание ракушек-рапан, чтобы самые выдающиеся экземпляры привести домой и положить начало коллекции. Мне отводилась скромная роль: нырнуть, разглядеть скопление этих моллюсков и указать их расположение, чтобы Ю.Л. на ощупь их мог достать. Но тут был один нюанс: я не могла правильно и быстро уходить под воду. Тогда он купил накануне бельевую веревку, взял хозяйскую камеру от колеса Камаза и мы молча двинулись в нашу бухточку. Я ни о чем не догадывалась, пока он не прихватил по дороге увесистый камень. Когда мы были уже далеко от берега, он привязал камень к веревке и скомандовал погружение. Я не сомневалась в честности его намерений, поэтому послушно, а, главное, быстро ушла под воду. Затем наступал второй акт придуманного им сценария. Посреди улицы устанавливались два кирпича наподобие таганка, водружалось помятое ведро с водой и начиналась операция по извлечению тела моллюска. Я наблюдала со стороны, как усаживались вокруг него босоногие ребята, с которыми он делился морскими трофеями, а также бездомные собаки-любители этого лакомства.

На следующий год мы были оснащены маской с очками и подводным ружьем, боевые качества которого Ю.Л. улучшил дополнительно тремя резинками. Мне была доверена только острога, но я получила право на автономное плавание. За следующие три года мы старательно исследовали подводный мир Крыма возле Гурзуфа, Артека, заповедника Никитского сада, уже после рождения старшего сына присоединили Крымское Приморье и Судак.

Эти экспедиции открывали все новые свойства его натуры. Он не воспринимал отдых ради отдыха и всегда был нацелен на какой-то результат, будь это на море или на даче. По этой причине он отвергал зимние занятия в бассейне.

В нем всегда жил охотничий азарт, который толкал его иногда на рискованные поступки.



Та самая скорпена

Однажды в погоне за большой рыбой он у меня на глазах прямо ввинтился в маленькую подводную пещеру и застрял плечами. Я понимала, что воздуха не хватает и ничем нельзя помочь, наконец, каким-то последним усилием он буквально выдернул себя из этого плена. И почувствовал себя победителем. В другой раз он наткнулся на ядовитый шип морского ерша-скорпены. Рука быстро начала опухать и болезненно пульсировать, отдавая нестерпимой болью во всем теле. На биостанции Никитского сада, куда мы обратились за помощью, сказали, что противоядия не существует, уж как повезет. Он промучился сутки, а затем все пошло по-старому. Единственная предосторожность заключалась в том, что он купил в аптеке маникюрные ножницы, привязал их к кукану на плавках и прямо в море, держа скорпену на острие стрелы, совершал обряд обрезания ядовитых шипов, а затем уже нанизывал рыбу на кукан. Рыбная охота имела еще и ту подоплеку, что мы с сыном обеспечивались обедом и ужином и были избавлены от стояния в очередях местной столовой, тем более, что качество нашей еды было несопоставимо.

Именно из Крыма пришла в нашу семью традиция приправлять рыбные блюда и даже пельмени соусом «Соломон», которому нас обучили местные рыбаки. После рождения младшего сына Александра мы редко отдыхали на море всей семьей, и, когда Саша подрос, обучение меня морскому делу было передано сыну (старший Сергей лето проводил в стройотрядах). Под упорным давлением сына я научилась прыгать с волнореза вниз головой и проныривать волну. Передача навыков от отца к сыновьям, приобщение их к «взрослому» делу всегда проходили не нарочито, а как-то само собою. Ю.Л. не упускал возможности взять с собой Сережу на научные школы-семинары, и он не только имел возможность увидеть Прибалтику, Байкал, Алушту, но и расширить свой кругозор, слушая доклады и общаясь с серьезными учеными. У всех мужчин в нашей семье было много общего не только в предпочтениях, но и в манере мышления, в способе изложения материала. Это было видно даже по их школьным сочинениям. Они были привержены, как отец, принципам логической строгости, ясности и простоте стиля. Не менее важной в «генетическом наследии» была основательность в подходе к разработке проблемы, независимо от темы, у Сергея - на поприще химической науки, у Александра в области программирования. Когда Александр оказался соавтором отца в нескольких монографиях, то его главы оценивались именно с этих позиций и Ю.Л., будучи очень придирчивым редактором, говорил мне, что ему почти не приходится вносить правку в текст работы Александра. Мы считали себя «избалованными» родителями в том смысле, что не знали проблемных забот об их учебе, но Ю.Л. хорошо представлял круг их интересов. Если старший к окончанию школы видел себя только химиком, то младший уже с пятого класса тянулся к профессии отца. Наконец, в восьмом классе он получил в подарок компьютер Commodore, что для 80-х гг. было еще большой редкостью. Почти весь класс прошел «школу молодого бойца» - программиста у нас дома.

Научная карьера Ю.Л. и прямо и косвенно влияла на семейный быт. Для меня эти перемены обозначались новыми темами и терминами в разговоре, появлением его новых публикаций, научными командировками. В 1963 г. он приступил к чтению общего курса по программированию на кафедре прикладной математики факультета ВМК. На следующий год состоялось открытие НИИ ПМК и Ю.Л. предстояло сделать выбор между ВЦ и ПМК.

С ВЦ связывали годы плодотворной работы вместе с группой программистов, которые стали нашими общими друзьями. Этот выбор дался нелегко. Отдел, куда Ю.Л. был принят в качестве младшего научного сотрудника, возглавил С.И. Альбер. К этому времени кандидатская диссертация Ю.Л. была готова, но он непозволительно долго откладывал ее формальное завершение. При активной настойчивости заведомо он довел дело до конца и в 1965 г. успешно ее защитил. Одним из оппонентов выступил Э.З. Любимский.

Конец 60-х и начало 70-х мне запомнились как бум в развитии программирования, судя по косвенным, домашним наблюдениям. Во-первых, растущее количество перфокарт, с которыми он не расставался даже дома; во-вторых, значение слова «ошибка», оно преследовали его и ночью. Альбер мне как-то сказал, что умение Ю.Л. найти ошибку в программе сравнимо с умением найти иголку в стоге сена.

Затем началась бэйсикомания со своим слэнгом типа «ликбэйсик». Ю.Л. работал над первой в стране диалоговой системой программирования на базе языка BASIC. Он пишет учебник по программированию, который будет еще долго востребован самыми разными специалистами. Дома мне безуспешно пытались втолковать, что значит создать интерактивный компонент, повышающий быстроедействие рабочей программы. Кстати, это было сделано намного раньше, чем за рубежом. Если бы в это горячее время можно было спросить психолога о состоянии Ю.Л., он, вероятно, назвал что-нибудь вроде «программистской эйфории». Однажды, находясь в таком настроении, он мне предложил: «Давай я научу тебя программировать. Я могу научить даже фонарный столб!» Надо было согласиться, хотя бы на элементарную компьютерную грамотность.

В его архиве хранится весьма пухлая папка с просьбами выслать систему программирования БЭЙСИК. Все адреса перечислять слишком долго, удивляет их география: Москва, Саратов, Казахстан, Нижний и область, Шахты, Тула, Новосибирск, Владивосток, Сухуми... Точка в научных и романтических его отношениях с Бейсик на протяжении 20 лет была поставлена в итоговой работе «Толковый словарь языка программирования Бейсик», изданной в 1992 г. и которая по сути представляла собой энциклопедию терминов, используемых в Бейсик-системах программирования. В том же году им была защищена и докторская диссертация на тему «Создание инструментальных программных средств для разработки диалоговых систем САПР и АСНИ». Подготовка к ней не заняла много времени, ибо стала обобщением многолетнего опыта работы над расширением диалоговых возможностей языков программирования и внедрения его в практику крупных промышленных предприятий.

Насколько Ю.Л. был однолюбом в своей специальности, настолько разнообразен он был в своих увлечениях-хобби, причем к ним подходил с характерной для него системностью, не свойственной дилетантам. Если говорить о нем как о грибнике, то необходимо упомянуть и подбор литературы о грибах, в том числе на польском и чешском. Он не был знаком с этими языками, но, обучаясь в украинской школе, и, зная болгарский, обладая незаурядным лингвистическим чутьем, он легко разбирался в тексте. Недаром к нему обращались за помощью юристы с просьбой переводить документы с болгарского и украинского языков. Теоретические экскурсии в мир грибов дополнялись грибными фото-сессиями, ибо без фотоаппарата он в лес не ходил. В результате, проска-

нировав свои снимки и соединив их с полученными через Интернет, он стал обладателем удивительной по красоте коллекции, которой часто любовались его гости.

Походы за грибами начинались в мае, когда появлялись первые строчки и сморчки, и кончались в сентябре. Эта охота в компании с Геннадием Григорьевичем Денисовым готовилась и как серьезная экспедиция и как приключение. Отправлялись затемно, дожидаясь рассвета у костра, чтобы первыми войти в лес. Грибной инструмент в виде аккуратно отшлифованной палки-рогатки (до сих пор лежат в кладовке) и «тормозка» из необходимых в таком случае продуктов и напитков готовились заранее. Это было не столько добывание грибов, хотя для них было делом чести принести полную корзину, сколько наслаждение жизнью. Случалось, ему приходилось реально рисковать здоровьем, а то и жизнью во время грибных походов. Неудивительно, что он мог оказаться в ситуации, когда его выручали только сила характера, природная выносливость и смекалка. Однажды в лесу он порвал Ахиллово сухожилие на правой ноге, боль сопоставима с огнестрельным ранением. Он не только дошел до электрички сам, но и не бросил корзину с грибами.

За это пришлось заплатить далеко раздвинутыми концами сухожилия, что заметно осложнило операцию. В другой раз его подвел глазомер, он принял болото за зеленую поляну и прыгнул с разбега. Трясина быстро начала затягивать и доходила уже до груди, а никакой опоры не было, кроме пустой корзины, а в ней фотоаппарата. Усилием воли он заставил себя хладнокровно, мелкими рывками приблизиться к суше и ухватиться за ветку. Когда он мог уже полагать себя спасенным, он сначала сфотографировал это место (снимок хранится в семейном архиве) и только потом выжал воду и в мокрой одежде еще плутал, отыскивая дорогу к станции. Это случилось в последние дни сентября, но все обошлось даже без насморка. Уверенность и даже самоуверенность, способность сохранять спокойствие и точно просчитывать алгоритм своих действий не раз выручали его в самых непредвиденных обстоятельствах и выявляли особенности характера.



Зелёный город, август 1961 г.

Невольно задумаешься, откуда у него такая любовь к природе среднерусской полосы? («Откуда у парня испанская грусть?») Родился он в Иркутске, куда его отец Лазарь Иванович Кетков вынужден был уехать не по своей воле, оставив работу в Киеве. Он занимал довольно высокий пост в структуре украинского комсомола, занимаясь работой по организации пионерского движения. В свое время он получил хорошее гуманитарное образование на болгарском факультете при Одесском университете. Там же и познакомился с будущей женой Ольгой Самойловной Фрум, которая заканчивала агрономический факультет. Когда в 30-е годы начались репрессии и аресты, он уже был в Ямаровке, в 50 км от Иркутска вместе с женой и сыном Ремом. Перед войной семья вернулась на Украину, но уже в родные края Лазаря Ивановича, в Запорожье. Вскоре он ушел на фронт, а семья была эвакуирована за Урал. Там, в поселке Кувандык Оренбургской области им пришлось пережить первую самую голодную зиму. Там же Юля пошел в школу, сразу во второй класс. Лазарь Иванович был ранен при переходе через Сиваш и после госпиталя присоединился к семье. Но после войны дороги родителей разошлись. Юля жил то с мамой в Горьком, где их приютила ее сестра Фрима Самойловна, жена профессора И.М. Коренмана, вместе с многочисленными родственниками (квартира на ул. Минина 5 оказалась так плотно заселена, что Юлю формально прописал к себе сосед академик А.А. Андронов), то с отцом. Но большую часть этих трудных послевоенных

лет он провел на Украине в болгарском селе Андровке, в доме своего дяди Ильи Дмитриевича Червенко. Мне посчастливилось неоднократно бывать в их доме во время летнего отпуска. Илья Дмитриевич был личностью незаурядной во всех отношениях. Директор школы, прирожденный педагог, он профессионально следовал принципам Макаренко и пользовался непререкаемым авторитетом не только среди односельчан, об этом свидетельствует его деятельность в качестве депутата Верховного Совета СССР. Илья Дмитриевич никогда не упоминал о своей награде - ордене Ленина, мы об этом узнали случайно. Этой семье, где его считали сыном наряду с другими четырьмя детьми, Ю.Л. был обязан воспитанием в нем многих замечательных качеств, прежде всего трудолюбия, представлению о порядочности, долге и сострадании.



И.Д. Червенко

«Двукорневая» система интеллектуальной природы Ю.Л. обеспечила добрые всходы. Проучившись на Украине почти до 8 класса, он закончил восьмую школу в Горьком с серебряной медалью. По его словам, он недолго размышлял над выбором специальности. Документы подавал на радиофак Горьковского университета, но, видя, что опаздывает на поезд на Украину и не остается времени на заполнение 5-и страничной анкеты (обязательной для данного факультета), перенес свои бумаги в комиссию физико-математического факультета. Жизнь показала, что выбор был сделан правильно. Учеба ему всегда давалась

легко и получение красного диплома стало естественным и логичным ее завершением.

Необычным оказался их курс выпуска 1957 г. Мало можно найти примеров такой студенческой дружбы, когда сокурсники регулярно на протяжении не одного десятка лет собираются вместе, вспоминают, выпускают газеты, коллажи, шаржи, обмениваются фотографиями детей, внуков, а затем и правнуков. Ю.Л. бессменно возглавлял организационную кампанию. Он не только держал в своем банке данных все старые и новые адреса, ответы на приглашения, но и лично посещал некоторых сокурсников в Нижнем, чтобы уточнить адрес. Последний сбор состоялся в 2012 г., ему физически нелегко было браться за это дело, но он не мог изменить своему долгу.

При всем обаянии компанейского человека, он отличался сдержанностью и всегда отстаивал свое право на личное пространство, куда допускались немногие. В нем доминировало мужское начало, что не исключало душевной ранимости. Такое сочетание всегда находило отклик в его окружении. Он отвергал любые посягательства на его личную свободу или попытку измерять его общей линейкой. Его дух бунтовал против физической слабости. Когда возраст стал настойчиво напоминать о себе, он старался не снижать темпы работы и страдал от вынужденных перерывов. До конца дней он не утратил способности наслаждаться работой любого формата. Через его руки прошли самые разнообразные домашние предметы - от сломанных кранов до обитых им диванов. Если нельзя было подобрать шкаф нужного ему размера, он делал чертеж и сооружал свою модель. Ему было в удовольствие поставить забор и другие необходимые постройки на даче, предварительно обработав каждую доску горбыля вручную. И он не мог смириться, когда тяжелая болезнь в 2006 году обрекла его на постельный режим. Тогда он, полулежа, подготовил к изданию свой курс «Введение в языки программирования С и С++», которые появились в издательстве Интернет-университет информационных технологий в том же году.

Представление о личности Ю.Л. будет неполным, если не упомянуть о месте юмора в его жизни. Начать с того, что с детства ему сопутствовал потрепанный том Рабле «Гаргантюа и Пантагрюэль». Он был передан сыновьям с рук на руки, затем в домашней библиотеке заняли свое место Ильф и Петров, Гашек и сборник «Габровские анекдоты», причем тексты жили активной жизнью, цитировались и перечитывались. Шестилетний Саша во время экскурсии с отцом в Суздаль, к удовольствию пассажиров, успел пересказать все габровские анекдоты. Ю.Л. любил подмечать комичные ситуации и ценил остроумие.

В его архиве хранится записная книжка с записью экспромтов советских писателей, сделанных во время второго съезда и попавших на стенд в кулуарах. Как Ю.Л. получил доступ на съезд писателей и к этому стенду – неизвестно, но их оригинальность неоспорима, хотя они и не были опубликованы.

Ю.Л. и самому было свойственно спонтанное, неожиданное остроумие. Помню, как он с моим братом Володей Фадеевым хохотали, придумывая отдельные реплики для истфиловского театра. Он хорошо чувствовал слово и мог извлечь совершенно неожиданный смысл, изменяющий объем его значения. Когда на 16 странице «Литературной газеты» появилась рубрика со словами-перевертышами, он стал отсылать в редакцию свои удачные находки.

Можно с уверенностью сказать, что и творчество Высоцкого оказалось так близко Ю.Л. не только свободой высказывания, но и едкой насмешкой над лицемерием и парадоксами нашей общественной жизни в конце 70-х годов. С появления первых записей певца он начал собирать их со страстью коллекционера. У нас дома «тусовались» поклонники и единомышленники опального певца, сопоставлялись разные варианты текстов, уточнялись факты авторской принадлежности тех или иных песен, анализировались иносказания. Задолго до официальных публикаций произведений Высоцкого друга Ю.Л. имели на руках весьма полный сборник, отпечатанный Ю.Л. на компьютере и собственноручно переплетенный (переплет книг также его хобби).

В 2000-х годах ситуация имела продолжение с песнями Тимура Шаова, который представлен в его коллекции в самом полном виде. Последней частью музыкальной коллекции Ю.Л. стало собрание русских романсов. Начало положил его корреспондент из Канады. Этот любитель и знаток русской музыкальной культуры активно ее пропагандировал, предлагая слушателям многообразный спектр исполнительства старинных романсов, читая лекции о биографиях и судьбах артистов. Став обладателем такого богатства, Ю.Л. серьезно увлекся, стал подбирать библиотеку, следить за выходом новых записей. Ему хорошо работалось под музыку городского и цыганского романсов.

Природная одаренность и вера в неисчерпаемость своих сил побуждали его пренебрегать осторожностью вопреки своему физическому состоянию. Он и скончался на ходу, по дороге на экзамен, в вестибюле Университета, где проработал почти 56 лет.

И.В. Кеткова

Мой друг Юля Кетков

Его уход я ощутил как большую личную потерю. Мало было людей за пределами семейного круга, которых я так любил – а теперь осталось еще меньше.

Познакомился я с ним, конечно, в НИИ ПМК. До сих пор помню, как он безвылазно сидел в своем маленьком кабинете на втором этаже, сразу налево от лестницы.

Главное Юлино свойство – невероятная отзывчивость, постоянная готовность помочь всем и каждому. Первый раз я посетил его дом (еще на улице Сусловой), приведя к нему для консультации своего дальнего родственника, мелкого бизнесмена из Богородска. Тот пытался помочь какой-то школе с компьютерной техникой (централизованных поставок тогда еще не было). Юля показал ему какой-то домашний компьютер, надавал много советов. Впрочем, последствий этот визит не имел. В другой раз я привел к нему (уже на Студенческую) свою студентку – тоже за какой-то консультацией. У Юли тогда даже не возникло вопроса – надо ли это ему...

При всей доброжелательности, Юля не был «добреньким», когда доброта служит просто маской для равнодушия. Когда было надо, он умел быть жестким и принципиальным. Так он фактически закрыл дорогу для защиты докторской диссертации одного моего знакомого, которая была интересна по содержанию, но слишком небрежно написана. Зная о моих близких отношениях с автором, Юля показал мне и диссертацию, и свой негативный отзыв. У меня даже возникла мысль: не отредактировать ли текст диссертации, но я не решился предложить свои услуги...

Еще один аналогичный случай известен мне опосредовано: из файлов в его компьютере, который Юля мне подарил, не озаботясь чистой памяти (что я воспринял как знак высокого доверия) – там небрежность оформления была лишь мелкой добавкой к научной недобросовестности автора, о которой Юля и написал в своем отзыве без всякой дипломатии, «открытым текстом».

Целая эпоха в нашей с ним жизни – школьные и студенческие олимпиады по программированию и информатике. Небольшой группой (3-4 человека) мы по многу раз встречались, готовили задания для «олимпийцев». Потом была сама олимпиада, разбор задач для учителей и для участников, проверка работ, тут уже больше людей было задействовано. Юля неизменно с раннего утра до самого конца «играл первую скрипку»: ходил по терминал-классам, общался с «олимпийцами». Словом, это была ЕГО олимпиада.

Важнейший сюжет – Ю.Л. Кетков и программирование. Я всегда был энтузиастом-любителем и, естественно, не мог обходиться без Юлиной помощи. Давным-давно, в эпоху программирования в кодах ЭВМ М-20, он сделал мне щедрый подарок – нумерованный экземпляр книги Ляшенко, тогда вышедшей

только для служебного пользования (в «нормальном виде» она была издана гораздо позже). Кто помнит те времена, оценит щедрость подарка.

Потом я с его помощью осваивал Бейсик и Фортран (Паскаль, правда, выучил сам).

Где-то в начале, как сейчас принято говорить, «нулевых годов» Юля сделал мне предложение – заняться Матлабом (это комбинация из своеобразного языка программирования и почти необозримого количества прикладных программ, поставляемых фирмой MathWorks). И не просто заняться, а принять участие в написании руководства.

Я, конечно, согласился, вспомнив совет замечательного математика А.Я. Хинчина: «Если хочешь узнать какой-то предмет, возмись его преподавать» (злые языки перефразировали: «если не знаешь...»). Разделение труда было очевидно: Юля писал о программировании, я о математических методах (впрочем, все написанное мной, Юля тщательно прочитал и «приложил свою руку»). Еще один соавтор – младший Юлин сын Саша (в его студенческие годы я вел у него практику по курсу Геометрия и Алгебра). Саша писал о сложных вопросах компьютерной графики. В 2004 г. наша книга вышла в издательстве БХВ-Петербург, годом позже, когда появилась новая версия Матлаба, вышло слегка переработанное издание.

Естественно, в процессе написания книги мы общались очень часто, я постоянно бывал у него дома. Когда книга была уже «на выходе», издатели захотели поместить на обложку фотографии авторов. В этот момент я оказался болен, поэтому Юля пришел с фотоаппаратом ко мне домой, а может быть в больницу – сейчас точно не помню.

Именно в этот период более плотного «домашнего» общения я познакомился с очаровательной Юлиной женой Идой. Как-то так сложилось, что мы с ней сразу стали «на ты», поэтому я даже толком не знаю ее отчества. Сейчас мы с ней иногда перезваниваемся ...

М.М. Шульц

Старший преподаватель кафедры математической логики и высшей алгебры
факультета ВМК ННГУ

Слово о друге и коллеге

С Юлием Лазаревичем Кетковым мы познакомились как-то незаметно. Я пришёл работать в Вычислительный центр Горьковского государственного университета в 1969 г., и, конечно, слышал о нём, как об уникальном программисте.

Мы начали общаться на постоянной основе, когда Юлий Лазаревич во главе коллектива разработчиков создавал первый транслятор с Бейсика для машин типа М-20. В 1972 г. в Вычислительном центре Горьковского государственного университета был создан первый в СССР интерактивный терминал-класс на восемь пультов на базе ЭВМ М-222, и для учебных целей транслятор был немного модернизирован для работы в таком многоуровневом варианте. Операционная система М-222 не предполагала использовать полную память машины – 8 кубов. Очень большая по тем временам память, 4000 слов в каждом кубе. Задача была в том, как использовать ее целиком, все было рассчитано на 4096 слов, а у нас их было 32000. В конечном итоге Юлий Лазаревич программную часть задачи, а мы – ее аппаратную часть, решили.

Параллельно началась работа в университете по развитию компьютерной геометрии и графики. Эта тема была основой и кандидатской, и докторской диссертаций Юлия Лазаревича. Он принимал участие в работах по системе «Чертёж». Это система, которую разрабатывал Институт кибернетики Академии наук Украины для систем БЭСМ-6. Тогда у нас пошли с ним более плотные контакты, поскольку оба занимались похожими задачами. Было различие в направлениях, поскольку у нас были разные заказчики. В 4 отделе НИИ Прикладной математики и кибернетики, где он работал с 1965 г., по заказу Военно-морского флота, осуществлялось проектирование подводных лодок. В рамках этой системы была решена задача одновременного вывода графической информации сразу на несколько дисплеев с тем, чтобы показать конструкцию в трёх проекциях и аксонометрии. Поскольку программно это сделать не удавалось, в Киеве сделали аппаратно: перепаяли БЭСМ-6 под большие экраны, поставили спецоборудование. И четыре изображения объекта выводились так, как должны были выглядеть на чертежах. Вот это была система «Чертёж». И вот, решая эту задачу, мы быстро перешли на «ты», а вскоре стали большими друзьями, несмотря на двенадцать лет разницы.

У нас было много точек соприкосновения по работе, и вспомнить можно многое. Но самые сильные у меня воспоминания о Юлие Лазаревиче, как о человеке. Отличительной чертой его характера была доброжелательность. Если кто-то обращался к нему с просьбой или вопросом, он никогда не отмахивался, ссылаясь на занятость. Он всегда помогал найти ответ. Юлий Лазаревич был большим эрудитом и мог легко сопоставлять несопоставимые с точки зрения

других людей вещи. У него было великолепно поставлено перо – так хороши были его статьи и работы.

В 1953 г. был создан Институт прикладной математики АН СССР, которым руководил тогда президент Академии наук СССР М.В. Келдыш. Институт был образован для решения сложных математических проблем, связанных с государственными программами исследования космического пространства, развития атомной и термоядерной энергетики, на основе создания и широкого использования вычислительной техники и программного обеспечения.

Когда институт получил первые машины, они были «голые», и Юлий Лазаревич занимался разработкой для них программного обеспечения. В ИПМ была крепкая команда программистов, работавших с БЭСМ-6. Именно на созданном ими программном обеспечении были решены задачи, связанные с атомной тематикой – в частности, с моделированием атомных взрывов. И очень многие задачи космической программы, навигации баллистических и крылатых ракет.

Эти же работы были использованы в НИИ ПМК, когда решалась задача навигации наших крылатых ракет – полет крылатой ракеты по заданной траектории над заданной местностью, чтобы ракета пришла в нужную точку без отклонения. Уже тогда при решении этой задачи закладывались основы современных геоинформационных систем, таких, как GPS и ГЛОНАСС. Частным случаем этой сложной задачи стали работы по картографии, которую в НИИ ПМК вёл и ведёт профессор Ю.Г. Васин. Отдел Юлия Лазаревича занимался разработкой картографического ПО, и насколько высоким был уровень разработки, говорит один интересный случай: с 1977 г. в России стали проходить сначала Всесоюзные, а затем с 1990 г. Международные конференции по компьютерной графике – Графикон. В 1994, 2001 и 2002 гг. Графикон проходил в Нижнем Новгороде на базе Нижегородского государственного университета им. Н.И. Лобачевского. Я и Юлий Лазаревич Кетков были председателями конференции. В 1994 г. на Графикон приехал профессор Дэвид Арнольд из Университета Восточной Англии. Мы сумели организовать ему экскурсию в НИИ ПМК и стали показывать работы по картографии. В этот момент наш слабенький компьютер принимал со спутника пакет информации. Пока мы разговаривали, машина выдала полный анализ с рассчитанной картой, за десять минут. Английский гость был в шоке – в Англии на подобный результат требуется три-четыре дня, хотя машины там намного более мощные. «Нам такой результат не по зубам», – честно признался он.

Для меня очень важной была наша совместная работа в Совете по защите докторских и кандидатских диссертаций, который я возглавляю. Мнение Юлия Лазаревича было для всего Совета одним из критериев объективности. Если какая-либо из выдвинутых диссертаций получала у него отрицательную характеристику, после рассмотрения мы всегда убеждались, что это заслуженная

оценка. И, что характерно, ни один из соискателей не вступал с ним в полемику, стараясь доказать обратное.



Графикон-2006

Юлий Лазаревич отстаивал основные этические принципы классического академического образования: любая диссертация пишется самостоятельно и должна отвечать всем требованиям Российской системы аттестации научных кадров. Эта система не менялась сто лет, но переход нашей страны на экономические рельсы серьезно повлиял на незыблемость её принципов. Особенно это стало заметно в гуманитарных науках. Юлий Лазаревич считал это большой проблемой.

Могу сказать, что и на заседаниях Государственной экзаменационной комиссии ННГУ, где я являюсь председателем, Юлий Лазаревич ставил высокую планку для студентов. По его мнению, студент, который защищает диплом, уже является специалистом и, соответственно, не может рассчитывать на послабления. Но все его замечания были объективны и справедливы, студенты никогда не протестовали против его решений.

В конце добавлю о машине ГИФТИ – первой созданной в нашем городе электронно-вычислительной машине. Машина ГИФТИ представляла собой

универсальную ЦВМ последовательного действия с оперативной памятью из 2016 слов длиной по 32 бита. Специально для нее в ОКБМ был изготовлен магнитный барабан, вращавшийся со скоростью 6000 об./мин. На этом барабане помимо ячеек оперативной памяти были реализованы сверхбыстрые рециркуляционные регистры, позволившие довести скорость работы арифметического устройства до 6000 сложений в сек. В арифметическом устройстве машины ГИФТИ командой программистов, в числе которых был Юлий Лазаревич, была реализована схема ускоренного умножения и смоделирована оригинальная схема ускоренного деления двоичных чисел. В составе этой машины был один из первых в мире (если не первый) графических растровых дисплеев. А знаете, что впервые было на нём показано? Новогодняя ёлочка и шутивная просьба программистов о повышении зарплаты. Программу, конечно же, писал Юлий Лазаревич.

У наших машин могло быть прекрасное будущее, но, к сожалению, у руководства страны было на этот счет свое мнение. Решение перейти на копирование зарубежных аналогов принималось первыми лицами страны. А ведь БЭСМ-6 была, к примеру, уникальной машиной – ее КПД (с операционной системой, написанной при участии Юлия Лазаревича, составлял 96%! Только 4% ресурсов шло на собственные нужды ОС. А у IBM S/360 КПД составлял 60-64%. Наша БЭСМ-6 решала 256 задач на одном процессоре одновременно, и на каждую задачу был свой выносной пульт. Машину отличала колоссальная надежность, сбой случался один раз в полтора-два года. Между прочим, последнюю в Вычислительном Центре ННГУ БЭСМ-6 демонтировали в 1994 г.

Я считаю, что в Вычислительном центре ННГУ была великолепная команда. Надеюсь, что следующие поколения программистов, воспитанных такими уникальными людьми, как Юлий Лазаревич Кетков, будут держать планку, и традиции нашей школы останутся с ними на долгие годы.

С.И. Ротков

Заведующий кафедрой инженерной геометрии,
компьютерной графики и автоматизированного проектирования ННГАСУ,
доктор техн. наук, профессор

Человек с большим сердцем

Наш отдел Математического обеспечения ЭВМ долгие годы (с 1965 г.) является не только трудолюбивым, трудоспособным коллективом, но и настоящей крепкой семьей, где все знают обо всех, где всегда готовы прийти на помощь, где радость одного члена коллектива становится радостью всех его членов.

И конечно, главная заслуга таких теплых взаимоотношений в отделе - заслуга нашего руководителя Юлия Лазаревича Кеткова, который был душой коллектива. Юлий Лазаревич – Человек с большой буквы, находивший время все видеть, везде успевать, всюду творить добро.

Приведу один пример... Шел конец 1999 г. В Чечне с новой силой разгорались боевые действия. У моей сестры Галины Дмитриевны (представьте – не у меня) там были дочь с тремя детьми и сноха с двумя ребятишками: Мадина, Залина, Хава, Аминат и Хадижат. Сначала эти молодые женщины с малолетними детьми спасались от войны в Ингушетии, но с наступлением холодов, не имея средств и теплой одежды, приехали в Нижний Новгород. Кетков и тут проявил свои лучшие качества.

Узнав о случившемся, он предложил коллективу, помочь, кто, чем может. И уже на следующий день несколько мешков теплой одежды не только для детей, но и для взрослых, были в отделе. Этот «ценный» груз незамедлительно был отправлен адресату. Сколько неопишуемой радости было и у взрослых, и у детей. Еще бы – суровая зима теперь была не страшна. Но на этом шефство Юлия Лазаревича не закончилось. Зная, как дети любят сладкое, он не раз привозил сладости детишкам. Ну и конечно, без подарков к Новому году и к 8 Марта никто не оставался.

После того, как я вышла на пенсию, виделась с коллективом реже, но наше общение с Ю.Л. Кетковым осуществлялась по телефону. Он находил несколько минут в своем плотном графике, звонил, и мы душевно беседовали. Его интересовало всё. И в разговоре он всегда передавал приветы и поздравления с праздниками моей сестре и интересовался делами тех «беженцев», которые уже стали студентами вузов.

Но и «беженцы» с благодарностью вспоминают этого человека с большим сердцем.

Молоткова Т.Д.

работала в 4-м отделе НИИ ПМК в 1965-1999 гг.

Учитель с большой буквы

В жизни каждого человека есть люди, которых хочется называть Учителями, у меня первый Учитель - это Юлий Лазаревич Кетков. Даже не верится, что о нем надо говорить в прошедшем времени.

Общение с ним мне дало очень многое в то время, когда мы были просто студентами, и предположить о наших дальнейших успехах было тогда сложно.

К середине учебы на факультете ВМК нас уже успели помучить адресным программированием в кодах, и это занятие не всех прельстило. И тут в расписании мы видим спецкурс - «Современное программирование». Мне это показалось интересным, и я стал посещать этот спецкурс. И вместо восьмеричных кодов команд и адресов ячеек прозвучали волшебные для нас слова: Фортран, Алгол, Бейсик... Это был уже какой-то другой мир. Спецкурс увлек, я ходил на него с удовольствием, с трудом, как и все студенты, запоминая фамилию и имя-отчество лектора. Подошел экзамен, я сдал его на 5, и на подведении итогов экзамена услышал слова, меня весьма удивившие: «А Вас, молодой человек, я попрошу остаться».

И я с удивлением узнал, что лектор вовсе не доцент университета, а заведующий отделом научно-исследовательского института Прикладной математики и кибернетики, и ему сейчас нужны люди на работу, в том числе и студенты. И мне предлагается начать работу в этом отделе. И на оставленной мне записке с телефоном я прочитал «Кетков Юлий Лазаревич».

Бурлила наша студенческая жизнь: самодеятельность, стройотряды, турпоходы и т.п., но предложение очень заинтересовало. Я собрал необходимые документы после зимней сессии и отправился в НИИ ПМК. Недолгое оформление, и вот у меня уже свой рабочий стол. А за соседним столом сидит преподаватель, с которым за время учебы отношения складывались не самым лучшим образом.

Помню, меня быстро ввели в курс дела, я освоил новые языки и технику, и работа закипела... В скором времени понадобились еще руки, и с моей подачи в отделе стали работать и Саша Гребенников, и Володя Коротков, и Женя Кустов, да всех я сейчас и не припомню.

Отдел быстро оценил наше присутствие, и на столь дефицитное машинное время вечерами и в выходные дни отправлять уже стали нас, оставляя подробные инструкции и колоды перфокарт. С Юлием Лазаревичем мы общались практически каждый день, поскольку не всегда на печать выходили колонки цифр, да и полученные результаты требовали проверки. Он готов был отлаживать программы с каждым из нас, но при этом и учил этой тонкой науке.

Мы были одними из первых профессиональных программистов, и многое из

того, что мы делали, было впервые. Так, выполняя одно из заданий, я работал со стандартной программой для решения систем линейных алгебраических уравнений сотого и выше порядков. Сначала все пошло нормально, но контрольный просчет вручную показал возникающую при повторных вычислениях серьезную ошибку. Долгое время мы перепроверяли мою программу, но это ничего не меняло. И тогда я стал анализировать стандартную программу и нашел в ней ошибку! Это было для меня и большим успехом, и большим продвижением в нашей программистской науке.

С этой программой связано еще одно воспоминание. После занятий мы с Сашей Гребенниковым как всегда ехали на работу, и в троллейбусе увидели профессора, который только что читал нам лекцию по физике. К нашему удивлению он тоже вышел на площади Минина и направился к институту. Мы начали обычный рабочий день, и вдруг меня вызывают к Юлию Лазаревичу. Я захожу к нему в кабинет и вижу там нашего профессора, а Юлий Лазаревич просит меня помочь профессору с работой с той злополучной программой! Мы проходим в лабораторию, я вникаю в задачу, все подробно объясняю, но вижу на лице профессора замешательство. В конце концов, он не выдерживает и спрашивает: «А Вы с утра не были у меня на лекции?» После моего утвердительного ответа профессор сказал: «Да, к своим студентам я на консультации еще не ходил...» Конечно, у профессора программа заработала, и он получил необходимые результаты, о чем он напомнил мне на экзамене...

Мы стали полноценными членами коллектива, принимали участие во всех неформальных мероприятиях отдела, где видели Юлия Лазаревича и как открытого, веселого человека. Кстати, в традиционном КВНе между преподавателями и студентами он был капитаном команды преподавателей, а я - студентов. Наши шутки типа: «Если из вашей зарплаты вычесть нашу стипендию, то еще и на трамвай останется», – помогли нам выиграть, а назавтра я шел на работу с некоторым чувством неловкости, но за обедом Юлий Лазаревич еще раз посмеялся над прошедшей игрой...

Примерно в таком же ключе прошел у меня и дипломный проект, который тоже был отражением выполняемых заданий, но снова не сходились цифры. Подняв сопромат, который мы не изучали, перед самой защитой я нашел ошибку не в своей программе, а в техническом задании, основанном на трудах известных ученых, о чем Юлий Лазаревич доложил ГЭК. К моему изумлению, эту историю через 40 лет я услышал от Валерия Николаевича Шевченко.

Университет окончен, но меня ждет не продолжение работы в отделе, а служба в армии. Приехав в отпуск, я пришел в НИИ ПМК, пообщался со своими соратниками, поговорил с Юлием Лазаревичем, и он дал мне письменную характеристику, где четко и красноречиво описал меня как специалиста. Эта характеристика очень помогла мне в служебном продвижении и в поступлении

в адъюнктуру. На защите я показал результаты сложнейшего моделирования на ЭВМ и стал кандидатом технических наук по космической навигации. Конечно, без опыта работы в НИИ ПМК такого результата достичь было бы сложно.

При каждом приезде в Нижний Новгород мы общались с Юлием Лазаревичем, он был в курсе моих дел и от души радовался моим успехам. Работая преподавателем в военном училище, я принял участие в разработке, аналог которой был описан в статье Андрея Петровича Ершова в середине 1980-х годов и стал основой школьного курса информатики. Я попросил Юлиа Лазаревича помочь мне попасть к А.П. Ершову, и оказалось, что там работает однокурсник Кеткова Юрий Абрамович Первин. **В скором времени я в Новосибирске в академическом городке общался с А.П. Ершовым.** Наша работа была им высоко оценена, потом встречались еще несколько раз, мы получили рекомендации к публикациям в несколько журналов, в том числе и только что появившемся «Информатика и образование». Эти рекомендации и новый круг знакомых помогли мне стать одним из первых докторов педагогических наук по специальности «Теория и методика обучения и воспитания (информатика).

Последние годы мы встречались с Юлием Лазаревичем в университете, куда я приезжал на заседание диссертационного совета. Он был энергичен, полон планов... Для меня он навсегда останется Учителем.

У меня более 400 публикаций, я подготовил около 40 кандидатов и докторов наук, сейчас работаю заместителем директора по инновациям Федерального государственного научного учреждения «Институт информатизации образования» Российской академии образования, Заслуженный работник высшей школы РФ в области образования, профессор, полковник в отставке. Я всегда буду помнить человека, помогавшего идти к этим успехам со студенческой скамьи.

О.А. Козлов,
Выпускник ВМК 1972 г.

Газета «Нижегородский университет»,
№ 07, август 2014 г.

Тот самый Кетков!

В институте Технической кибернетики АН Белоруссии, где я был в командировке, меня как-то спросили:

- А кто Ваш научный руководитель?
- Кетков Юлий Лазаревич.
- Тот самый Кетков?!
- Да, **ТОТ САМЫЙ!!!**

В большинстве вычислительных центров СССР Кеткова знали – кто-то учился по его учебникам, кто-то использовал его программные продукты, кто-то слушал его доклады на конференциях. Поэтому подобный вопрос я слышал не раз, благо, что фамилия Кетков достаточно редкая.

Перефразируя поговорку «Водопадов много, а Ниагара одна», можно смело сказать: «Программистов много, а Кетков ОДИН!»

А начиналось мое знакомство с **ТЕМ САМЫМ КЕТКОВЫМ** так... На ВМК я поступил в 1969 г. К программированию у меня с самого первого курса было особое отношение. Удивительное ощущение испытываешь, подчиняя своей воле самое сложное инженерное и научное творение, когда-либо созданное человечеством. Я с упоением писал и отлаживал простенькие программы.

Лишь много позже я узнал знаменитые слово академика А.П. Ершова: «Программист должен обладать фантазией Эдисона и скрупулезностью бухгалтера, чтобы сооружать все, что угодно из 0 и 1».

На первом курсе мы программировали в кодах ЭВМ М-20. Огромная ЭВМ занимала половину первого этажа на физфаке на улице Свердлова. Для всех своих знакомых я организовывал экскурсию. Это был 1969 г., когда слова: ЭВМ, космонавтика, кибернетика, ядерная физика – были овеяны необычайной романтикой. Среди своих друзей и знакомых гуманитариев благодаря этому я имел очень высокий статус.

На втором курсе программирования у нас не было, и вновь появилось во втором семестре третьего курса. Я ждал этой встречи с любимым предметом «как ждет любовник молодой минуты верного свидания».

И вот эта встреча – в аудиторию буквально влетает запыхавшийся Кетков и своим изумительным каллиграфическим почерком пишет на доске непонятные символы - АЛГОЛ-60. И дальше пошел рассказ о самом распространенном в то время в СССР алгоритмическом языке.

К середине лекции мое внутреннее возмущение достигло максимума: а где мои любимые коды М-20, которые я так обожал?! О них Кетков не говорил ни слова. И только к концу лекции я стал понимать, что программирование - это нечто большее, чем я представлял на первом курсе.

Потом были АВТОКОД и БЭЙСИК...

К середине весеннего семестра третьего курса нужно было выбирать кафедру и научного руководителя для курсовой работы и дальнейшей специализации. Мы с моим приятелем Андреем Сорокиным решили идти к Кеткову, тем более, кто-то из однокурсников сказал, что у Кеткова решают реальные задачи, которые непосредственно внедряются в практику.

После одной из лекций два самонадеянных третьекурсника, полагающих, что осчастливят доцента своим предложением, подошли к Кеткову и попросили взять их для выполнения курсовой работы. Ответ был совершенно ошарашивающий: Кетков, деликатно извинившись, твердо сказал, что очень занят по основной работе и курсовиков и дипломников не берет.

Нас это обескуражило и пришлось искать других руководителей. Моя курсовая было связана с дифференциальными уравнениями, к которым я никогда не испытывал особой любви.

Прошел год... Мой приятель Андрей предложил снова подойти к Кеткову, уж очень нам хотелось писать программы. Наша любовь к программированию существенно усилилась, после того, как мы получили на третьем курсе пятерки на экзамене у Кеткова. Он никому не ставил двойку, но получить пятерку было очень сложно. Разрешал во время экзамена пользоваться любой литературой и выходить из аудитории. Он досконально вникал в написанные во время экзамена программы, задавая множество вопросов, стараясь всегда быть предельно объективным. Из-за этого его экзамены заканчивались порой в 10-11 часов вечера. Обычно на группу из 25 студентов приходилось всего 2-3 пятерки.

Поэтому мы считали себя молодыми гениальными специалистами, жаждущими произвести научный переворот в программировании.

«Но ведь Кетков никого не берет из-за занятости», – напомнил я Сорокину.

«А мы его попробуем обхитрить. У Кеткова родился второй сын, и папаша пребывает в состоянии необыкновенного счастья», – сказал с лукавой улыбкой Андрей.

После лекции мы подкараулили Кеткова и вновь изложили ему свою просьбу. Он с улыбкой сказал, что помнит нас, но студентов по-прежнему не берет. На это мы хором заявили, что он через год обещал нас обязательно взять. «Неужели так и сказал?» – улыбнувшись, спросил Кетков. «Именно так и было», – хором ответили мы. «Ну, раз обещал – приходите завтра в НИИ ПМК»

Первый шаг в большое программирование был сделан: мы попали к знаменитому программисту – Юлию Лазаревичу Кеткову. Дело осталось за малым: сделать научное открытие. На следующий день в кабинете Кеткова на ул. Ульянова 10 мы получили задание – разработать алгоритмы функционирования системы с разделением времени на базе ЭВМ М-222. «Разделение времени – новое очень перспективное направление», – сказал Шеф. «А что можно почитать на эту тему?» – спросил я. «Публикаций еще очень мало. Американцы пыта-

ются что-то сделать, но пока еще на уровне опытных образцов», – иронично улыбнувшись, сказал Кетков.



Ю.Л. Кетков и А.Н. Рябов

Гордо выпятив грудь, озадаченные, мы вышли из его кабинета, абсолютно уверенные в том, что именно нам поручена историческая миссия – выполняя курсовую работу, не только догнать, но перегнать Америку и тем самым выдвинуть программирование в СССР на самые передовые в мире позиции.

Как часто мне потом уже в зрелом возрасте не хватало той иллюзорной юношеской самоуверенности, но крылья, которые стали расти в тот момент, благодаря моему новому руководителю, я ощущаю до сих пор.

Не откладывая дело в долгий ящик, мы нашли в университете свободную аудиторию и до поздней ночи, пока нас не выгнал вахтер, рисовали блок-схемы и алгоритмы. На протяжении двух месяцев мы не только все вечера посвящали курсовой работе, но и частенько пропускали лекции. Мы были уверены, что вплотную приблизились к американцам в создании систем с разделением времени. Стараясь не отвлекать Кеткова от его основной работы, мы усердно трудились самостоятельно и в середине мая понесли наш готовый опус на суд научного руководителя.

Увидев нас, он спросил, куда мы пропали на такое долгое время, на что мы уверенно сообщили, что все сделали самостоятельно. «Все?» – скинув брови, удивленно переспросил Кетков. «Все! Абсолютно все. Проблема решена, осталось только все запрограммировать», – заявили мы. «А посмотреть дадите?» – с улыбкой спросил доцент. «Конечно», – гордо ответили мы, протягивая папку с блок-схемами. Полистав в течение 10 минут наше творение, Кетков сказал, что по четверке мы заработали. Мне показалось, что я ослышался, так как был уверен, что за нашу работу просто было необходимо поставить пять с плюсом. «Сдавайте весеннюю сессию и в сентябре на пятом курсе приходите на практику. Займемся этим делом всерьез». «Как всерьез?! – подумал я. – Два месяца не ходили в кино. Все отдано большой науке!»

Спустя пять лет, опубликовав несколько работ на эту тему в научных изданиях, начиная понимать, что же такое программирование, я случайно узнал, что та курсовая работа имела «огромный успех» у сотрудников отдела математического обеспечения ЭВМ, в котором я потом проработал почти 15 лет.

После ухода «новоявленных гениев» Кетков собрал импровизированный семинар сотрудников отдела и зачитал пару страниц из нашего «научного трактата». Опытные программисты хохотали до слез. Бред, который они слышали, естественно ничего общего с наукой не имел.

Этот эпизод подчеркивает удивительный такт Кеткова. Он с уважением отнесся к нашему труду, не промелькнуло и тени превосходства над нами. Понял, что работа написана самостоятельно. А чтобы не убить наше рвение, желание творить и созидать самостоятельно, не унижил нас ни взглядом, ни словом, отнесся к нам как к равным, как к коллегам и предложил работать над задачей дальше.

Он понимал, что в молодом специалисте важно не погасить ту искру жажды познания, которая поможет ему в дальнейшем. Как свечку прикрывают ладошкой, чтобы она не погасла, так и мой Учитель всегда заботливо оберегал меня, а я этого и не замечал. Как заботливая мать с волнением и радостью наблюдает первые шаги своего дитя, так и мой Учитель, понимая, что я еще не раз набью шишки в работе, переживал за меня, оберегал, но никогда не ограничивал в самостоятельности и свободе.

Рябов А.Н.

директор издательства «Нижегородская летопись»,
кандидат физ.-мат. наук,
работал в 4-м отделе НИИ ПМК в 1973-1988 гг.

Воспоминания об отце

В детстве я рос в окружении компьютеров, и они не казались мне чем-то особенным, хотя и вызывали жгучее любопытство. Года в четыре я уже побывал за ЕС'овским терминалом ЕС-7920 и, хотя от основной ЭВК его от греха подальше отключили, я всласть понажимал на клавиши и научился стирать набранные символы BackSpace'ом. К пяти годам я отвечал всем взрослым, что когда вырасту, обязательно стану системным программистом. Это не совсем укладывалось в распространённую парадигму «комонавт-лётчик-пожарный», поэтому некоторые спрашивали: почему именно системным? И я, насколько мог и понимал сам, старательно объяснял разницу между системным программированием и разработкой прикладного ПО.

Думаю, во многом это стало следствием того, что отец никогда не делал секрета из своих занятий, никогда не уходил от ответов на вопросы, говоря, как многие взрослые: «подрастёшь—поймёшь», а старался объяснять сложные вещи из мира вычислительной техники простым языком, доступным и четырёхлетнему ребёнку. Он учил меня переводить десятичные числа в двоичную систему и обратно, писать программы на Бейсике, рассказывал про ассемблер и машинные коды. Когда классе в 5-м в моей школе задали за лето нарисовать план пришкольного участка с использованием картографических знаков, я задумал сделать это на каком-нибудь графопостроителе в НИИ ПМК. Пришлось осваивать Фортран, поскольку картографический пакет ГРАФОР был написан именно на нём. Задача для пятиклассника оказалась слишком сложной, я сделал её процентов на 60, но отец честно занимался вводом моих перфокарт и приносил мне промежуточные результаты, исчёрканные из-за ошибок с подъёмом/опусканием пера плоттера.

Мне несомненно повезло, что моим обучением занимался один из лучших специалистов в своей сфере, который не жалел ни времени, ни сил на то, чтобы я освоил науку программирования. В старших классах школы у нас дома появился первый персональный компьютер, Commodore PLUS/4 со встроенным Бейсик-интерпретатором, кассетным магнитофоном и современным на тот момент устройством внешней памяти – накопителем на гибких магнитных дисках. Может быть, именно наличие Бейсика повлияло на решение о покупке, хотя компьютерных магазинов тогда не было, куплен он был в «комиссионке» и особого выбора не было. Отец всю жизнь был равнодушен к Бейсику, несмотря на все его ограниченные возможности и пренебрежение со стороны профессиональных разработчиков. Зато для обучения программированию с самых малых лет Бейсик подходит прекрасно. Commodore PLUS/4 сегодня занял место в экспозиции музея факультета ВМК.

Отец всегда интересовался новыми реализациями Бейсика на различных компьютерах. Тогда в СССР количество образцов зарубежной вычислительной

техники было сильно ограничено, особенно, если речь шла не о компьютерах промышленного применения. Чтобы ознакомиться с отдельными образцами, приходилось ездить в разные организации, а иногда и преодолевать препятствия. Однажды Министерство внешней торговли СССР приобрело некоторое количество персональных компьютеров фирмы Yamaha для своих сотрудников, в которые тоже был встроен Бейсик-интерпретатор. Отец договорился с кем-то из знакомых, чтобы можно было приехать и «пощупать руками» эту конкретную реализацию. Он взял с собой меня, и мы столкнулись с совершенно неразрешимой ситуацией – Министерство внешней торговли находилось на Смоленской площади в Москве в одном здании с МИД СССР, а там был довольно строгий пропускной режим. Мне на тот момент ещё не исполнилось 16 лет, и паспорта у меня не было – выписать пропуск без паспорта возможным не представлялось. Однако люди, увлечённые компьютерами, всегда находят общий язык и при посредничестве таинственного персонажа, не то Семёныча, не то Степаныча, который среди вопросов эксплуатации здания на Смоленской площади отвечал и за запасы технического спирта, я смог преодолеть милицкий кордон на входе и попасть к заветным компьютерам.

Однажды отец привлёк меня к настоящей взрослой работе – нужно было написать программу ввода и обработки данных анкет для Горьковского института инженеров водного транспорта. Работу предстояло выполнить на ЭВМ «Искра-226», основным языком программирования для которого был, как можно, наверное, догадаться, достаточно мощный Бейсик.

Как-то раз, в одной организации списали несколько матричных принтеров Robotron с последовательным интерфейсом RS-232, которым обладал и наш Commodore, и один из них достался мне. Тогда, на первом курсе, под руководством отца я писал свои первые в жизни драйверы, программу для распечатки изображений, осваивал ассемблер. Программы для нашего компьютера были в ту пору большой редкостью, распространялись на магнитофонных кассетах, но отец нашёл в Горьком ещё одного любителя, у которого дома был Commodore 64 – родственник компьютер с частично совместимым кодом и мы обменивались программами. Я, конечно, не был знаком с теми задачами, которые решал отец на работе, но компьютерная графика появилась в моей жизни уже тогда и надолго определила сферу моих интересов и занятий.

Для меня никогда не стоял вопрос, куда пойти учиться после школы и чем заниматься в жизни, это было очевидно с самого детства. Я был студентом своего отца и посещал все его лекции и спецкурсы, причём отнюдь не из сыновнего долга – мне всегда было интересно на этих лекциях. Несмотря на то, что многие вещи отец уже успел мне рассказать раньше, или я мог расспросить его о чём угодно дома, его лекции всегда были очень интересными, я всё равно узнавал что-то новое. Единственным исключением были зачёты и экзамены –

отец никогда не принимал их у меня, всегда поручал оценку моих знаний другим, чтобы она была объективной.



Во время моей учёбы существовала традиция выпускать на День ВМК специальные деньги, они действовали только во время праздника и называли их в честь одного из преподавателей.

Все основные изменения в общественно-политическом устройстве страны как раз пришлось на время моей учёбы в университете. Так сложилось, что поступал я в ГГУ, а заканчивал уже ННГУ, наша самая повышенная стипендия на ВМК, сравнимая с зарплатой молодого специалиста на производстве (105 рублей получал Ленинский стипендиат), внезапно обесценилась. К третьему курсу мне пришлось найти себе работу, и она оказалась напрямую связана с компьютерной графикой, дизайном, вёрсткой, обработкой фото- и видеоданных. До 1999 г. я, так или иначе, был связан с компьютерной графикой, и во всех позднее написанных совместно с отцом монографиях у нас существовало «разделение труда» - я писал главы, посвящённые компьютерной графике, отвечал за программы-примеры и вёрстку книги, а основные главы писал отец. Мы писали и о конкретных реализациях языков программирования, и о специфических системах типа MATLAB, а последняя книга была посвящена СПО – программированию на Free Pascal'e.

Отцу посчастливилось увидеть своими глазами эволюцию вычислительных систем от механических арифмометров и первых ламповых монстров до современных супер- и микрокомпьютеров. Для каждого из компьютерных поколений он писал программы и учил это делать других. Он учил не только программированию как ремеслу, но искусству изящного решения сложных задач, любви к знаниям и интересу ко всему новому. Знания должны получаться самостоятельно в ходе исследований и экспериментов, а программное обеспечение должно быть эффективным, отлаженным и документированным, считал Юлий Лазаревич.

А.Ю. Кетков

Небожитель

Придерживаясь сухого «документального» стиля, следовало бы начать с того, что Юлий Лазаревич Кетков – это руководитель моих курсовых, дипломной работы, а также моей кандидатской диссертации. Однако, обдумывая этот текст, я постепенно осознал, что знакомство с этим неординарным Человеком, Педагогом и Профессионалом – совершенно не ограничивалось, как часто бывает в годы учебы, эпизодическими обсуждениями текста диплома, постановкой задачи, и даже не сводилось к 4-5 гг. учебы в аспирантуре. Я вдруг осознал, что был знаком с Юлием Лазаревичем почти 14 лет. Все началось с моих первых дней учебы в Нижегородском государственном университете, это знакомство повлияло самым замечательным образом не только на факт овладения специальностью и появления ученой степени, но и на всю мою профессиональную карьеру, даже на сферы жизни, не относящиеся к работе и учебе.

Вспоминаю первый курс ВМК, «Прикладная математика», 1999 г. У нас было много неординарных лекторов – с одной стороны «мэтры» – Дмитрий Владимирович Баландин (я с ним был знаком по «ученической научной работе»), Сергей Николаевич Слугин, Николай Яковлевич Коган, многие другие и «молодёжь» – Николай Юрьевич Золотых, Майданов, Алексева. Юлий Лазаревич Кетков – он один – самым удивительным образом объединял в себе качества и мэтров, и молодежи!

Посудите сами: насколько я помню, он никогда не приходил на лекцию в костюме. Запомнилась его слегка пружинящая походка, кожаный коричневый портфель и, кажется, тогда кожаная куртка – не современная мода, но отчетливо отсылавшая к молодежи или даже к советскому кино про лётчиков. Во время лекции – постоянный контакт со слушателем, шутки, совсем юношеская мимика; и тут же – веское слово, четкое, неторопливое, но полное энергии. Любил он встать одной ногой на ступеньку кафедры, другой опираясь на пол, в пол-оборота к аудитории – совсем по-юношески, по-пиратски даже. За всей этой «энергичной» вывеской студентам давался идеально выверенный (казалось столетиями!) материал – и, кстати, совершенно замечательный почерк, аккуратный даже на доске.

В литературе описаны – да и в жизни каждого студента известны случаи, когда средних лет педагог «заигрывает» с молодёжью. Известны и примеры преподавателей, которые «несут себя» и всегда «держат дистанцию». Здесь ничего подобного не было! Да, авторитет Юлия Лазаревича был, можно сказать, колоссален – это с одной стороны. Но, как я со временем понял, Кетков был всегда очень энергичен, молод и демократичен в душе, по характеру, по жизненному настрою, и в 1990-е – 2000-е гг. это, разумеется, проявлялось еще больше.

В общем, неудивительно, что как опытный и «молодой лектор, которому было далеко за 60», Юлий Лазаревич снискал себе заслуженную славу абсолютно у каждого студента, кому довелось слушать его «Методы Программирования». Все это подогревалось фактом, не замалчиваемым в стенах ВМК, что Кетков – был «первый программист Горького, а то и СССР». Слава Кеткова среди студентов была вещь само собой разумеющейся, даже несмотря на то, что наше поколение уже не училось по его книгам, подобно тому, как это делали студенты лет за 10 до нас.

Вспоминаю знаковый для меня случай – на первом курсе Ю.Л. Кетков устроил мини-соревнование. «Кто напишет самую компактную программу для игры в крестики-нолики, тому бонус – на экзамене», – по-моему, так это прозвучало на лекции в тот день. Тут случилось первое из череды «совпадений», которых потом в нашем знакомстве с Юлием Лазаревичем было необыкновенно много. Совпало два факта: во-первых, на нашем потоке набралось много «сильных программистов», и они попросту вздернули нос: после олимпиадных задачек эта задача была «унизительно простой». С другой стороны, у меня, на 1-м курсе никак не претендовавшего на роль «опытного разработчика программного обеспечения», в памяти всплыло, что когда мне было лет 10-11, вероятно, отлеживаясь дома после гриппа, я написал маленькую «книжицу» со «стратегией»/алгоритмом такого рода простой игры. Так и совпало, что я взялся за эту задачку. В итоге исходный код моей «программы» по количеству символов «выиграл» (чисто условно за счет чрезмерной грубости «подхода»). Победа (плюс условная слава среди непрограммирующей части факультета) была за мной, ну, а Кеткову только того и надо: основная масса студентов не спит, элемент соревнования и практики внесен в лекции, отдельные студенты отбираются на будущую специализацию.

После этого Юлий Лазаревич, кажется, предложил мне что-то сделать более серьезное, но я пошел на попятную – играла роль загруженность и робость первокурсника. В общем, до самого последнего момента я не думал идти к Юлию Лазаревичу на специализацию; отчасти я воспринимал Кеткова как своеобразного «небожителя» (несмотря на всю условную «молодежную» манеру общения), а отчасти я, как и многие по неопытности, тянулся к молодым педагогам. Было мне немного неловко и за «задний ход» после истории с крестиками-ноликами.

И здесь позволю себе сконцентрироваться на теме «совпадений», о первом из которых уже сказано выше. Перечислю только лишь самые «знаковые» из них. Научной темой Ю.Л. Кеткова оказалась цифровая картография. Моим (наверное, редким) увлечением в школьные годы – была география с картографией в том числе, поэтому я и оказался в итоге среди дипломников Юлия Лазаревича.

Небольшой трехэтажный дом на 20-30 квартир по ул. Студенческой, где жили Кетковы, оказался тем же домом, где много лет прожила моя мама в детстве. Для полутора миллионного города – интересное совпадение.

Кстати, моя прабабушка Елена Иродионовна Стрелкова, директор школы (очень повлиявшая на всю учебную и профессиональную судьбу моей мамы) была знакома с Кетковыми; и много лет назад, когда родители Иды Владимировны Кетковой подыскивали место за городом, продала Кетковым свой садовый участок.

Сестра моей бабушки – по другому «совпадению» – училась на смежном курсе с Кетковым и одно время дружила с той небольшой компанией, к которой относился и Юлий Лазаревич.

Надо сказать, что все эти фамильные «совпадения» выяснились далеко не сразу, иные пару лет спустя после того, как я стал появляться в доме на ул. Студенческой.

Юлий Лазаревич поспособствовал моему устройству на первое место работы. Собственно, совпадение заключалось в том, что Ю.Л. Кетков по ошибке устроил меня «не туда», что сильно повлияло (как я считаю очень положительно) на мою дальнейшую карьеру.

Дело было так. В 2003 г. весь мой курс проходил собеседование с сотрудниками «Телмы», вроде бы у меня все прошло хорошо, но отбор шел на обучение в УЦ «Телмы», а не на трудоустройство. Мне хотелось уже работать, о чем я вскользь упомянул Кеткову (возможно, похвастал), а он сообщил, что знаком с директором этой фирмы и на всякий случай спросит его о вакансиях. Сказано – сделано, и примерно через 2-3 недели я шел в «Телму» на собеседование; курьез был лишь в том, что Юлий Лазаревич направил меня в «Теком», а не в «Телму» (путаница, возможно, вышла из-за созвучия в названиях). В итоге «собеседовался», а затем трудоустроился я в компанию «Теком», где на тот момент было куда больше возможностей для творческого развития профессиональных навыков, чем в более «конвейерной» атмосфере «Телмы-Телеки».

Еще одно, так скажем, рутинное совпадение – Кетков, кажется, был одним из первых на факультете ВМК, кто поучаствовал в первом совместном микрогранте между отделом «инструментов оптимизации» нижегородского «Интелла»; в этом отделе на сегодняшний день я и работаю.

Приведены примеры самых «ярких совпадений», при внимательном рассмотрении их было куда больше.

Хочется сказать несколько слов и о стиле работы Юлия Лазаревича со студентами и аспирантами. Прежде всего Юлий Лазаревич был очень аккуратен в самых разных смыслах этого слова. Это касалось и «стиля программирования», документирования исходного кода. Это же касалось и его

подхода и его требований к оформлению текста диплома, диссертации. Юлий Лазаревич без усталости старался «привить» мне культуру оформления научного текста, как четкими рекомендациями, так и собственным примером.

Еще один штрих к портрету: и дома, и на работе в НИИ ПМК у Юлия Лазаревича хранилось много тысяч перфокарт (еще с тех легендарных времен большой ЭВМ на Покровке...), которые он использовал для того, чтобы формулировать постановку задачи, ТЗ, формировать список дел для студентов.

Нельзя не упомянуть и того, что «обучение примером» иногда переходило в иной разряд: случалось, что Кетков, видя мои не слишком удачные попытки написания отдельного фрагмента статьи или диссертации, приносил мне на следующей неделе практически готовый текст к тому или иному разделу. Такое происходило нечасто, может быть, раз пять за все годы, но это были именно те случаи, когда наступал «критический момент». Низкий поклон за это Юлию Лазаревичу. Такой пример действовал сильнее доброго десятка советов или подсказок; есть в этом, на мой взгляд, и проявление высокого профессионализма и, одновременно, альтруизма, резко контрастирующего с практиками иных нынешних научных руководителей.

Нельзя сказать, что мой путь к защите кандидатской диссертации был очень быстрым, но вот когда текст диссертации был практически готов, Юлий Лазаревич (неожиданно для меня) проявил удивительную энергичность. За несколько недель он успел организовать предзащиту, найти хорошие издания для дополнительных публикаций, начал активно работать со мной по целому ряду вопросов оформления.

Процесс не двигался слишком гладко, так как научный совет на ВМК ННГУ в тот момент в большей степени подходил для «остепенения» будущих кандидатов физ.-мат. наук, особенно это стало понятно во время предзащиты. Но и здесь Юлий Лазаревич проявил удивительную энергию и точность расчета: в какой-то момент он сообщил мне, что защищаться будем в Строительной академии. В этом решении, как я понял позднее, было сразу несколько положительных моментов: во-первых, это был один из немногих Ученых советов по компьютерной графике и инженерной геометрии в стране (а тема моей работы напрямую относилась к этой области), во-вторых, речь здесь шла, естественно, о степени кандидата технических наук. В целом, в состав совета входили очень сильные профессионалы.

В этой перестановке оказался и принципиальный момент для Кеткова. Правда, узнал я об этом лишь в день защиты в стенах ННГАСУ. Вспоминаю этот июньский день: я как раз окончил свое выступление, настало время отзыва научного руководителя; за столом сидят профессора, специалисты по «computer science», проявившие себя еще в советское время; были, конечно, и молодые ученые. Юлий Лазаревич начал свое выступление с очень добрых,

положительных слов в мой адрес, он всегда очень активно вступался и поддерживал своих учеников на защитах, что тоже не часто встретишь сегодня. В какой-то момент Кетков стал плавно переводить тему в область характеристики программной инженерии в советских и российских реалиях. В нескольких словах он обрисовал сложные условия, когда зачастую, имея доступ к меньшим ресурсам и менее мощной аппаратуре, советские, а затем российские программисты, обеспечивали лучший по производительности и эффективности результат, чем их коллеги за рубежом. Достигалось это за счет применения более сложных и эффективных алгоритмов, технических решений; не удивительно, что в области прикладной инженерии и алгоритмистики в нашей стране были достигнуты замечательные результаты.

Здесь Юлий Лазаревич перешел ко второму аспекту «неравенства» отечественных специалистов. Несмотря на огромные усилия и вложения в области «computer science», советские, а затем российские Ученые советы никогда не воспринимали работы с инженерным, техническим уклоном соразмерными «научнообразным» физико-математическим проблемам. По словам Кеткова, первые Ученые советы по техническим алгоритмическим специальностям сталкивались с огромными трудностями; всё, что не казалось близким к фундаментальной науке, воспринималось критически. И это в то время, когда в ведущих лабораториях и университетах Запада по соответствующим специальностям уже полным ходом защищались молодые ученые-технари.

Это были замечательные, можно сказать программные слова, которые произвели очень большое впечатление на всех присутствующих. Я в тот момент вдруг осознал, почему все мои попытки изобрести, «создать» на пустом месте научнообразные описания, формулы никогда не воспринимались Юлием Лазаревичем с большим энтузиазмом; хотя, в то же время, он зачастую поощрял учеников к созданию четких, однозначно трактуемых описаний алгоритмов и оптимизационных техник.

Кетков закончил свое выступление словами о таланте отечественных специалистов и необходимости поддерживать кандидатские работы по техническим специальностям. И это прозвучало как настоящее кредо человека, который никогда не «отступал назад», если речь заходила о защите своей Профессии – Профессии с большой буквы.

В этом тексте я коснулся профессиональной и педагогической деятельности Юлия Лазаревича, затронул параллели с моей собственной биографией, но при этом я не сказал ни слова о той замечательной атмосфере, которую мельком наблюдал, заходя время от времени к Кетковым в гости на полчаса-час – в их замечательную квартиру на Студенческой улице.

Наверное, об этом лучше скажут и напишут те, кто был близко знаком с Юлием Лазаревичем «по жизни», а я только добавлю несколько штрихов.

Прежде всего, Кетков любил говорить о внуке и внучках, которых он, по-моему, очень сильно любил. И недаром он был так рад и даже немного похвастался, когда ему подарили электронную фотографическую рамку, где постоянно «крутилось» слайд-шоу с фотографиями маленькой Юли, а затем и Кати. Эти разговоры всегда переводились на расспросы о моих дочерях и, конечно, подобные беседы всегда были очень теплыми, душевными по тону.

Почти каждый раз, когда я заходил в двери квартиры Кетковых, меня встречал роскошный кот с лоснящейся длинной шерстью. Юлий Лазаревич в обязательном порядке ругал этого кота, часто подталкивая его подальше от прихожей, но при этом по интонациям было почему-то сразу понятно, что хозяин к своему коту просто очень сильно привязан!

Совсем понемногу, но достаточно интересные вещи говорил Юлий Лазаревич и о своих увлечениях, касаясь, например, любимых музыкальных исполнителей. А как-то раз зашел совсем уже неожиданный разговор о будущем России. Я совсем не ожидал, что мой научный руководитель вот так переведет учебную тему на вопросы политики. Но по ходу разговора стало понятно, что Юлия Лазаревича очень тревожит известная тема «нефтяной зависимости» и одновременно определенной технологической отсталости нашей страны. С осторожностью, хотя может быть и с некоторой надеждой, говорил он тогда про «Сколково».

Где-то за неделю до кончины Юлия Лазаревича, что-то меня будто бы сильно подтолкнуло, и я решил позвонить – поздравить своего бывшего научного руководителя с прошедшими новогодними праздниками, хотя, к своему стыду, последние два года я очень редко созванивался с ним. В телефонной трубке я услышал знакомый бодрый и полный энергии, хоть и постаревший голос; конечно же, мы коснулись темы внуков и дочек, говорил Юлий Лазаревич об олимпиадах и планах на новый учебный год. А уже спустя несколько дней его не стало...

Проходит время, уже забываются конкретные детали, но зато в памяти все четче формируется какой-то удивительный единый образ университетских лет, где воспоминания о годах, проведенных в университете, об однокурсниках, об аспирантских годах – нераздельно сливаются с обликом, с воспоминаниями о моем наставнике Кеткове Юлии Лазаревиче.

З.А. Матвеев
Выпускник факультета ВМК

Ю.Л. Кетков – пионер в Нижегородском программировании

Впервые я узнал о Юлии Лазаревиче в 1996 г. на первом курсе факультета ВМК ННГУ, где он преподавал дисциплину «Методы программирования» на языке Турбо/Борланд Паскаль. Позже я выбрал специализацию по кафедре МО ЭВМ и руководителем своей курсовой работы Кеткова Ю.Л. В то время он набирал только двух студентов. Моя работа была связана с доработкой языка описания графических изображений (расширением множества команд). После IV курса я поступил в магистратуру, и работа продолжилась ещё на 2 года, после чего Ю.Л. Кетков позвал меня работать в НИИ ПМК.

Первое, что бросилось в глаза на лекциях Ю.Л. Кеткова – это его почерк. Практически всё написанное на доске, - от текстов программ, до обычных комментариев, - было написано почти как на печатной машинке: идеально ровные горизонтальные строки и буквы одинаковых размеров. Поначалу я думал, что это делается исключительно для удобочитаемости, но потом, во время написания курсовых работ, убедился, что также идеально выглядели не только надписи на доске, но и черновики.

Важной отличительной чертой лекций Ю.Л. Кеткова являлась «их адаптивность». Помимо обязательных сведений по предмету, он иногда рассказывал о различных новостях в мире компьютерной техники и программирования, предупреждал о появлении новых вирусов, рассказывал некоторые интересные задачи из прошедших олимпиад по программированию. При всём при этом, он всегда очень переживал, если по каким-либо причинам та или иная лекция отменялась. Раньше это случалось чаще всего из-за всяких праздников, а в последнее время добавилась и другая причина – по болезни. Чтобы Юлий Лазаревич не пришёл на лекции по болезни, болезнь должна быть очень веской. Например, при перенесённом воспалении легких и операциях по удалению тромбов в ногах Ю.Л. Кетков достаточно скоро начинал ходить на лекции и на работу в НИИ ПМК, хотя это ему и доставляло заметные со стороны трудности.

Отдельных слов заслуживает способ сдачи экзамена Ю.Л. Кеткову. В отличие от большинства преподавателей, которые достаточно критично относятся к наличию у учащихся разного рода шпаргалок, Юлий Лазаревич наоборот разрешал приносить на экзамены, не только шпаргалки, но и книги. На удивленные вопросы по этому поводу он отвечал: «Пусть используют что угодно, когда готовят ответ на вопросы билета, но когда человек садится ко мне отвечать, я по ответам на дополнительные вопросы всегда смогу понять, разбирается ли он в предмете или просто механически списал ответ откуда-нибудь». При сда-

че экзамена обычно требуется написать простенькую программу (10-20 строк). Излюбленным приёмом проверки таких программ являлось указание на тот или иной участок текста и вопрос: «А это что такое?». И действительно, если человеку кто-то помог написать программу или он нашёл готовое решение, то он терялся, а вот если экзаменуемый сам написал программу, он может объяснить любую строку кода. Когда Юлий Лазаревич ставил кому-нибудь двойку, он немного расстраивался и, похоже, воспринимал этот факт как свою недоработку.

На ведение курсовых и дипломных работ у Ю.Л. Кеткова тоже был свой взгляд. Если другие преподаватели требовали от учащихся явки в определённые дни, как на работу, то Юлий Лазаревич, напротив, выдавал задание и предоставлял учащимся практически полную свободу действий. Это означает, что человек сам уже потом решал, когда ему нужно будет явиться и показать результаты своей работы. Это ужасно удобно для людей, умеющих планировать своё время, и ужасно вредно для неумеющих. Последние начинали писать диплом/курсовую в последний месяц, а то и неделю, до окончания семестра. В результате, доставляли неудобства не только самим себе, но и Юлию Лазаревичу, которому иногда в спешке, почти в день сдачи, приходилось писать рецензии и отзывы на курсовые или дипломы.

Кстати, и к выбору тем курсовых Юлий Лазаревич подходил достаточно демократично. Если пришедшие к нему студенты уже ясно представляли, чем хотят заниматься, то и тема курсовой подбиралась в соответствии с пожеланиями. При этом Юлий Лазаревич досконально пытался изучить эту тематику. Если же студент не определился с темой, то ему на выбор предлагался набор задач, решение которых, могло бы упростить нашу работу в НИИ ПМК. В основном это касалось отображения и редактирования дискретных знаков, а также других задач геоинформатики.

Вообще, Юлий Лазаревич всегда очень обстоятельно подходил к любой новой тематике или крупной покупке. В такие периоды на его столе появлялось большое количество различных заметок, а в компьютере куча сохранённых сайтов и книг по нужной теме. При всём этом он не считал зазорным дополнительно расспрашивать окружающих по интересующим вопросам. Аналогичным образом он относился и к жалобам на ошибки в нашем программном обеспечении, тщательно изучая вопрос, зачастую выявляя неверные входные данные. Благодаря этому, о многих ошибках люди предпочитали сообщать нам в обход Юлия Лазаревича. Дело в том, что за годы работы в НИИ ПМК сложилась традиция поставлять нам повреждённые данные (не соответствующие требованиям) с сообщением об ошибке в наших программах.

Подобный обстоятельный подход к каждой проблеме, конечно же, отнимал значительное время. Но благодаря этому, когда при написании программ по той или иной тематике мы сталкивались с какими-либо проблемами, мы всегда могли задать вопрос Юлию Лазаревичу и почти на 100% могли быть уверенными, что эта проблема им уже обдумана или даже решена. Например, когда мы создавали комплекс программ по морской тематике, Юлий Лазаревич нарисовал все мыслимые и немыслимые условные знаки в любых вариациях. Поэтому при написании кода обработки того или иного знака, мы по мере необходимости брали уже готовые изображения...

Особых слов заслуживает и писательский талант Юлия Лазаревича. Он практически стал главным редактором всех исходящих отчётов и статей нашего отдела, искусно заменяя тяжёлые или неудобочитаемые предложения лёгкими и более простыми. Очень поражала его способность прикинуться «деревенским лаптем» при оценке статьи или программы. Наверное, именно благодаря этому, его книги написаны понятным и доступным для всех языком. При оценке программ на первое место он всё же ставил функциональность и работоспособность, а интерфейс пользователя называл необязательным украшательством.

В работе на посту начальника IV отдела Юлий Лазаревич руководил мягко в том смысле, что всегда старался избегать чётко указывать, кто и что должен делать и за какое время. Указания его носили скорее рекомендательный характер: «надо бы вот что сделать...», «надо бы статью на такую-то тему написать». В плане внесения каких-либо дополнений к программам мы были свободны, но обращались к нему в случае спорных вопросов или при планировании серьёзных изменений. В частности, Юлий Лазаревич частенько вставал на нашу сторону, когда требовалось написать что-то совсем уж невообразимое.

Когда я устроился в НИИ ПМК в 2002 г., Юлий Лазаревич позвал меня помочь в проведении олимпиады 2003 г.: поначалу в качестве волонтера – следить за соблюдением правил и порядка в терминал классах, а в последующие годы – в качестве члена жюри. В те времена олимпиадные задачи и контрольные тесты для них придумывали сами члены жюри. Иной раз придумывалось больше задач, чем нужно, поэтому они проходили некий отбор. Важным критерием отбора было, как говорил Ю.Л. Кетков: «В задаче должна быть «изюминка». Под этой «изюминкой» понималось то, что задача должна иметь два способа решения: первый – что называется «в лоб», который обычно занимает много времени; второй – нетривиальный и в то же время, существенно увеличивающий скорость решения. Такие задачи и считаются «олимпиадными»: кто заметит хитрость в задаче – тот сможет набрать максимум баллов, а кто не заметит – набирает меньше максимума. При этом важным является то, что задача при решении не разделяет людей, натасканных на решение задач, и лю-

дей, обладающих только знаниями по программированию: догадаться или не догадаться до хитрого решения могут и те и другие! Конечно, среди недогадливых специально подготовленные, всё равно, набирали больше баллов, чем не обладающие такой подготовкой участники. Тут уже играют роль знания по использованию быстрых алгоритмов для решения рутинных подзадач (например, таких как сортировка, поиск и т.п.).

После принятия задачи к олимпиаде, производилась её «доводка до ума». Традиционно к задачам готовились 5 тестов, поэтому «доводка до ума» заключалась главным образом в удобстве проверки. Юлий Лазаревич считал, что 5 тестов достаточно, чтобы рассмотреть все особенности задачи. Главными критериями удобства проверки являются однозначность ответа программы и наглядность этого ответа. Задачи формулировались таким образом, чтобы на любой допустимый набор данных программа должна ответить единственным верным образом (иными словами: на любой тест правильный ответ один единственный). Наглядность означает, что в случае неверного ответа программы, мы могли буквально «на пальцах» показать участнику, в чём заключалась ошибка, и какие случаи участник пропустил при создании алгоритма решения. Нередки были случаи, когда для наглядности подготавливались дополнительные материалы: картинки, графики, схемы решения и т.п. Эти материалы тоже способствовали обеспечению прозрачности проверки решений.

Непринятые или недоработанные задачи Юлий Лазаревич сильно упрощал и использовал в студенческих олимпиадах. На мои вопросы: «Почему для студентов нужны более лёгкие задачи?» Юлий Лазаревич отвечал: «Студенты более ленивые и уже неспособны быстро мыслить и решать задачи наравне со школьниками». Помню, что меня это очень удивило, но Юлий Лазаревич показал статистику решения подобных задач студентами, которая чётко продемонстрировала, насколько студенты слабее школьников. Будем надеяться, что в будущем студенты поумнеют.

Сам процесс проведения областной олимпиады в те годы был следующим: на написание решения участникам давалось 4 часа на каждом из двух туров (по часу на каждую задачу), после происходила проверка решений. Проверка была очной: член жюри садился вместе с участником за один компьютер, и программа вручную запускалась на каждом тесте. Такой метод проверки занимал порядка 2 часов, когда проверяло 8-10 человек. Проверка всегда велась достаточно демократично. Если участник забыл поправить в программе 1-2 символа, то ему разрешалось это сделать в присутствии проверяющего. Всякие бывали случаи: кто-то случайно нажал при финальном сохранении не ту клавишу и сохранил вместе с программой лишний символ, кто-то в спешке закомментировал не ту строку и т.п. При этом Юлий Лазаревич часто давал вкратце исправления, оговариваясь, что мы судим не способность не нервничать на

олимпиаде, а способность решать задачи. В то же время его по этому поводу, похоже, мучили некоторые угрызения совести, т.к. кто-то попросит жюри об одолжении, а кто-то постесняется. Другим признаком демократии была проверка решений отсутствующих. Дело в том, что многие участники областных олимпиад приезжают издалека и поэтому вынуждены уезжать раньше. Такие решения мы проверяли заочно, и они нередко получали заслуженные ненулевые баллы.

После окончания каждого тура олимпиады мы дополнительно перепроверяли все решения дома. С началом второго тура мы обменивались данными перепроверки и выбирали максимальные баллы (за исключением случаев программ-пустышек или со случайными решениями). Оценки второго тура, к сожалению, таким образом уже не корректировались, данные перепроверки второго тура оставались у нас только для статистики.

В 2008-2009 гг. в корне изменились правила проведения областных олимпиад. Вкратце опишу эти нововведения: теперь одни и те же задачи присылаются из Москвы во все города, к каждой задаче прилагается по 20-50 тестовых данных. Можно сказать, что с 2009 г. и начался закат олимпиадного движения по программированию. Отношение Ю.Л. Кеткова к нововведениям было отрицательным, впрочем, такое отношение сформировалось у всех мало-мальски здравомыслящих людей. Дело тут даже не в том, что 20-50 тестов – это слишком много для того, чтобы проверить 1-3 особых случаев в решении задачи. Главным недостатком новых задач является то, что они совершенно бесхитростны, т.е. решаются «в лоб», как написано в учебниках. Таким образом, олимпиаду нового формата более правильно называть проверкой знаний информатики, а уж никак не олимпиадой. Причём таких задач с каждым годом всё больше и больше. В тестах внимание уделяется больше не творческому подходу к решению задач, а технике программирования. Новые тесты ловят не ошибки в алгоритмах, а ошибки в их кодировании: вдруг участник выбрал недостаточно большой тип данных, вдруг забыл о размерностях типов, вдруг не учёл размерность результата и т.п. Причём на таких вот мелочах участник теряет очень большое количество баллов.

Другим нюансом нового формата олимпиады стал огромный объём работы, которую требовалось произвести при проверке решений. На каждом туре участник решает по 4 задачи, каждое решение нужно проверить на 25-50 тестах и сравнить с правильным ответом. Естественно, ни о какой очной проверке речи уже идти не могло. Вдобавок задачи и тесты к ним были таковы, что даже объяснить участнику, почему его программа выдаёт неверный ответ, стало невозможным. Представьте тест размером в 1 Мегабайт и примерно такой же

ответ программы, которая в одном байте где-то ошиблась, - нереально глазами даже найти место ошибки, не то, что понять, в чём ошибка. Единственный выход - это автоматическая система проверки, которую господа из Москвы не поставляют, а лишь предлагают где-то купить.

Благо тексты задач и тестов к ним были известны за два месяца до начала. За это время я успел написать скрипты, автоматически запускающие программы участников на тестовых примерах и проверяющие правильность ответов. Набор написанных скриптов и составил базис системы автоматического тестирования Тот (по названию египетского бога).

Генезис программы под новыми и новыми требованиями к проведению олимпиады - это материал для отдельной книги. Скажу только, что эта корявая первая версия системы очень помогла нам хоть как-то провести олимпиаду 2008 г. В виду того, что решения проверялись заочно, было много «криков» как от учителей, так и от самих участников по поводу результатов.

Кстати говоря, именно эту первую версию системы тестирования Юлий Лазаревич и использовал для перепроверки решений дома.

В первые годы олимпиад нового формата задачи допускали простой метод проверки, т.е. сравнение ответа программы с правильным ответом. В последующие же начали часто появляться задачи, где на каждый тест можно было дать несколько правильных ответов. В 2014 г. таких задач было максимум 6 из 8. Именно подобные задачи составляют основную сложность при проверке – для каждой из них приходится писать отдельные программы верификации, которые отвечают на вопрос «правильно ли ответила программа на тест?». Программа верификации иногда по объёму и сложности превышает само решение раз в 10. Тут Юлий Лазаревич тоже проявлял свои демократические взгляды, никогда не навязывая свои способы проверки и даже не контролируя правильность моих программ.

В 2013 г. придумали новые препятствия для проведения олимпиады: теперь задачи и тесты к ним выдаются членам жюри за 4 дня до начала олимпиады. За эти 4 дня нужно успеть написать проверяющие программы, их отладить, проверить на всякие экстремальные условия работы. Ясное дело, это нереально. Как следствие, в 2013 г. тестирующая система дала первый сбой на одной задаче, из-за которой на нас и было много жалоб. Оказалось, что Москва сама не умеет проверять эту задачу (их алгоритм проверки считал их же решение неверным)!

Юлий Лазаревич каждый год писал свои замечания по поводу олимпиады «высшему начальству», где он перечислял все недостатки нового формата олимпиады, но всё это стабильно игнорировалось. Тем не менее, до самой смерти оставался бессменным председателем жюри. Когда 5 января 2014 г. мы встретились по поводу предстоящей олимпиады, Юлий Лазаревич сказал, что откажется от преподавательской деятельности, перейдёт на четверть ставки и будет заниматься только олимпиадами. К сожалению, этому не суждено было сбыться...

А.И. Кузнецов

работал в 4-м отделе НИИ ПМК в 2002-2014 гг.

Музей факультета ВМК ННГУ в образовательном процессе

*«Существует мнение,
что музей – это что-то
опрокинутое в прошлое...
На самом деле – это линза,
проецирующая будущее».*

Директор музея кинематографии
Наум Клейман

Введение

В 1963 г. в Нижегородском государственном университете им. Н.И. Лобачевского (ННГУ)¹ по инициативе профессора Ю.И. Неймарка был открыт факультет вычислительной математики и кибернетики (ВМК), который ныне является одним из ведущих учебно-научных подразделений университета. Профессор Юрий Исаакович Неймарк (1920–2011 гг.) – всемирно известный ученый в области технической кибернетики, ученик А.А. Андропова. Обладая блестящей интуицией инженера, Ю.И. Неймарк получил 24 авторских свидетельства по закрытым темам. Ю.И. Неймарк создал классический метод D-разбиений, успешно занимался разработкой центрифуг, имевших высокую степень технологического риска. Он один из первых начал систематическое изучение хаотических явлений в динамических системах: разработал теорию устойчивости не голономных механических систем, получил ряд фундаментальных результатов по динамике систем с сухим трением, имеющих применение в вибрационной забивке. Он создал научную школу по динамике, механике, автоматическому управлению и математическому моделированию, исследовал метод точечных отображений, ставший формой описания динамических систем, удобной как для изучения конкретных систем и численных исследований, так и для рассмотрения теоретических вопросов, уделял большое внимание просвещению, его математические традиции продолжают на факультете ВМК.

За годы существования факультета ВМК сформированы научные коллективы и школы в области математического моделирования, принятия решений и информационных технологий, получившие международное признание. В 2013 г. факультету ВМК исполнилось 50 лет. Музей факультета ВМК принял активное участие в юбилейных мероприятиях. В своей экспозиции он показал,

¹ С 2009 г. университет имеет статус национального исследовательского университета, в 2013 г. в числе 15 российских вузов он выиграл грант Правительства РФ для повышения международной конкурентоспособности и вхождения в ведущие мировые рейтинги.

что хранит научные, педагогические и культурные традиции факультета. История факультета – это теперь и история формирования музея, его коллекций, его миссии в деле воспитания студентов, их приобщения к научной и практической деятельности.

1. Создание и развитие музея ВМК

1.1. Цели и задачи музея

Накануне празднования 100-летия со дня рождения академика А.А. Андропова, ученые-математики ННГУ внесли предложение организовать научно-просветительное учреждение, посвященное истории факультета ВМК – музей. Не случайно создание музея было приобщено к этой юбилейной дате. Академик Александр Александрович Андронов (1901–1952) – советский физик, механик и математик, специалист в области электротехники, радиофизики и прикладной механики, продолжатель лучших традиций классиков естествознания, ученик и ближайший сотрудник Л.И. Мандельштама, основатель всемирно известной Горьковской школы теории нелинейных колебаний. В 1929 г. написал фундаментальную работу «Предельные циклы Пуанкаре и теория колебаний». С 1931 г. работал в Горьковском университете над новейшими проблемами теории нелинейных колебаний и теории автоматического регулирования, осуществил тесный контакт между физикой и математикой, реализовал новый подход в идее секущей Анри Пуанкаре и методе точечных отображений. По инициативе А.А. Андропова в 1945 г. был основан первый в нашей стране радиофизический факультет.

В 2004 г. ректор ННГУ Р.Г. Стронгин и декан факультета ВМК В.П. Савельев поставили цель перед сотрудницей музея, выпускницей радиофизического факультета Н.Г. Панкрашкиной: подготовить стационарную экспозицию. Были сформулированы задачи: 1) собрать образцы вычислительной техники, отражающие этапы развития ЭВМ, 2) отразить историю становления дисциплины «вычислительная математика» на факультете ВМК, 3) отразить в экспозиции результаты научной деятельности ученых и студентов ННГУ.

1.2. Становление музея

С первых дней музею сопутствовала удача: он получил мощную поддержку в лице преподавателей университета. Профессор Юлий Лазаревич Кетков в числе первых передал будущему музею образцы вычислительной техники (ВТ) из личной коллекции. Декан В.П. Савельев направил Н.Г. Панкрашкину в командировку в Москву с заданием найти документ Министерства образования об открытии в 1963 г. первого в нашей стране факультета ВМК. В

Центральном государственном архиве был выявлен приказ № 440 от 11 июля 1963 г., подписанный министром В.Н. Столетовым. Аналогичные факультеты в других вузах были открыты на несколько лет позже: в Москве ВМК (МГУ) – в 1970 г., в Казани ВМК (КГУ) – в 1978 г.

Параллельно со сбором экспонатов шла работа по выявлению устной истории факультета: состоялись встречи с основателем факультета Ю.И. Неймарком. Обсуждалась история кафедры «Теория управления и динамика машин» (ТУиДМ), его книга «Сухой остаток» [2].

Ю.И. Неймарк стал вдохновителем многих музейных начинаний, разработал стратегию развития музея, предусматривающую осуществление разнообразных программ, направленных на развитие интереса к созидательной профессиональной деятельности студентов и выпускников факультета. Ученики Ю.И. Неймарка доценты Н.Я. Коган и З.С. Баталова «вдохнули в музей душу», их воспоминания послужили основой для поисковой работы и экскурсий по истории научно-педагогической работы факультета. С.Г. Кузин – доцент кафедры «Интеллектуальные информационные системы и геоинформатика» – предложил концепцию экспозиции, в основу которой был положен вклад научных школ факультета в техническую кибернетику, вычислительную математику, информатику и компьютерные технологии. Предусматривались, например, темы «Счетные и электромеханические устройства», «Элементы вычислительной техники», рубрики «ВМК вчера» и «ВМК сегодня». Они должны были отразить историю факультета и его техническое оснащение, эволюцию информационных технологий и технических решений в различных сферах деятельности (государственное управление, здравоохранение, образование, бизнес, досуг и другие).

1.3. Раритеты музея

Формирование коллекции нашего музея имеет специфические особенности. Если в историческом времени раритетом становится предмет, которому не менее 50 лет, в компьютерном мире «возраст» экспоната сокращается на порядок. В техническом музее собирательская деятельность осложняется тем, что морально устаревшее «компьютерное железо» не успевает быть осмысленно как исторически ценное, оно подвержено скорой утилизации. В 1990-е гг. оно сдавалось, например, в утиль, как источник драгметаллов. Были утеряны прошитые нашим мастером-золотые руки В.Н. Гребенчиковым платы машины «Стрела», утрачена установка «Магнитный подвес Ю.И. Неймарка» [3].

Всеми своими богатствами, поступательным движением формирования нашего музея, поисками и открытиями мы обязаны сотрудникам и студентам университета: первой и основной является коллекция образцов вычислительной техники, фотографий и документов из собрания легендарного нижегородского

программиста профессора Ю.Л. Кеткова. Страсть к собирательству, бережное отношение к предметам и любовь к технике привели его к систематическому подбору экспонатов, которые Ю.Л. Кетков в течение нескольких десятилетий использовал на лекциях по истории науки. Профессор Юлий Лазаревич Кетков (1935–2014) – выдающийся нижегородский программист, компьютерщик и блестящий лектор по языкам программирования. Ю.Л. Кетков, будучи студентом ГГУ, дипломную работу делал у основателя советской кибернетики А.А. Ляпунова. После окончания университета Ю.Л. Кетков разрабатывал программы для тестирования машины ГИФТИ, в НИИ ПМК выполнял специальные задания Правительства, заведовал отделом математического обеспечения ЭВМ, разработал русскую версию интерпретатора языка BASIC, создал в ННГУ первую в нашей стране диалоговую систему коллективного пользования. Коллекция Ю.Л. Кеткова будет значительно расширена: сын ученого, А.Ю. Кетков, готовит к передаче в музей научно-педагогическое наследие, которое он называет с грустью: «134 кг железа и макулатуры моего отца».

В книге поступлений, в электронном ресурсе SAMIS зафиксированы 65 единиц хранения. Это блоки памяти, платы, элементная база ЭВМ всех поколений, персональные и мини-компьютеры (ПК «Commodore PUS/4», отечественный компьютер «Электроника 60» – раритеты нашего музея), магнитные ленты, интегральные микросхемы, электронные лампы, конспекты лекций по машине ГИФТИ и языкам программирования, техническая документация с грифом «Для служебного пользования» (гриф сейчас потерял значение секретности).

Для примера приведем научное описание одного из экспонатов: *арифмометр «Феликс» в Книге поступлений имеет номер МННГУвКП 3. Счетная машина производства завода «Счет.маши», город Курск, конец 1930-х гг., сделана из металла, имеет размер – 320x135x125 мм. Предназначение: девятиразрядный арифмометр для простейших арифметических действий: сложения, вычитания, умножения и деления. Заводской № 5168.*

Легенда. В 1964 г. данный экспонат перевезли в НИИ ПМК, на нем работал Ю.Л. Кетков. Сохранность полная. Дар Ю.Л. Кеткова.

Поначалу небольшие, но репрезентативные коллекции музея позволили организовать передвижные выставки на внешних площадках Нижнего Новгорода и Москвы, чтобы популяризовать идею музея. Были проведены выставки «От арифмометра – к персональному компьютеру» (музей «Нижегородская радиолaborатория»), «Дары и дарители» (музей истории ННГУ), «Звезда по имени Каплан» (астрономический институт им. П.К. Штенберга, МГУ). Мобильные выставки музея породили «цепную реакцию» даров, музейное собрание пополнилось девятью персональными коллекциями: 865 экспонатов. Список дарителей составили 59 человек – заведующие лабораториями, преподаватели,

студенты университета, в фондовом хранилище оказалось большое количество книг, фотографий как исторических, так и современных репортажных.

Открытие экспозиции было намечено к 45-летию факультета ВМК (2008 г.). Экспозиционные площади заняли коридор первого этажа главного корпуса ННГУ напротив лабораторий «Математические и программные технологии для современных компьютерных систем», «Высокопроизводительные компьютерные системы», «Центр компетенции высокопроизводительных вычислений на основе технологии Майкрософт», владеющих уникальным оборудованием, обеспеченным охраной, сигнализацией и видеонаблюдением [4]. Первая экспозиция включала витрины площадью 23,76 м². Первого декабря 2008 г. состоялась торжественная церемония открытия музея. Ректор ННГУ Е.В. Чупрунов и декан факультета В.П. Гергель символично разрезали красную ленточку, руководитель Центра прикладной информатики В.П. Савельев провел первую экскурсию. 1127 человек познакомились с музеем на традиционном празднике «День вычислителя».

1.4. Музей сегодня

В настоящее время стационарная экспозиция занимает площадь 43,76 м², в фондах хранится 14 персональных коллекций, всего около 6 тысяч экспонатов, из них около 300 уникальных чудом сохранившихся образцов вычислительной техники, документов и фотографий. Одной из тем экспозиции является «История кибернетики». Среди экспонатов – книги Норберта Винера, тяжелые блоки ЭВМ и персональные компьютеры (например, магнитный носитель ЭВМ СМ-4 со съемным диском емкостью 4 МБ, матричный принтер Star NX-1500, ПК «Роботрон», «Квант», «Корвет», «БК», отечественный графический манипулятор «Крыса»). С конструктивно сложной аппаратуры снят корпус. На зеркальной стенке шкафов отражаются составные части приборов. Для расширения экспозиционной площади используются презентации, электронные фоторамы. Совершенствуется текст экскурсии, рассказ начинается с первых компьютеров, для которых ввод программ и данных осуществлялось с перфокарт, использовался язык ассемблера, выполнялись операции над числами с плавающей запятой.²

Особый интерес вызывают экспонаты, истинные памятники истории науки и техники, такие как магнитная головка машины ГИФТИ, 1953 г. – уцелевшая техническая деталь одной из первых цифровых последовательных ЭВМ в СССР. Состоит из магнитопровода трансформатора диаметром 2 мм и головки диаметром 0,5 мм, даритель разработчик М.Я. Эйнгорин. Ячейка памяти машины М-20, 1964 г. Электронная лампа деформирована пожаром, случившимся в

² Историю компьютера и программирования начинаем с 1837 г., когда Ch. Babbage и A. Lovelace сделали проект «analytical engine».

недрах ЭВМ, даритель руководитель ВЦ ГГУ В.К. Матвеев. Учебный макет процессора на электронных лампах «15×15», 1967 г. – даритель зав. лабораторией А.П. Городилов. Ручной перфоратор для перфокарт, 1970 г. – даритель ст. преподаватель М.М. Шульц.

В музейной библиотеке имеются монографии, статьи, мемуары нижегородских корифеев-математиков, книги по программированию, ставшие библиографической редкостью [5]. Например, книга «Малая электронная счетная машина», авторы С.А. Лебедев (выдающийся разработчик вычислительных систем, уроженец Нижнего Новгорода), Л.Н. Лашевский, Е.А. Шкабара. Книгой пользовались инженеры НИИ ПМК, на ее обложке затерт гриф «совершенно секретно».

Экспозиция музея расширяется. В марте 2014 г. на кафедре ТУиДМ открыли два выставочных модуля, посвященных профессору Ю.И. Неймарку: «Жизненный путь» и «Выдающийся ученый». На открытии экспозиции ректор ННГУ Е.В. Чупрунов отметил, что музейное пространство позволяет проникнуть в персональную историю науки, президент ННГУ Р.Г. Стронгин говорил о незаурядном педагогическом таланте своего учителя Ю.И. Неймарка и его роли в области теории устойчивости неголономных механических систем.

Продолжается формирование коллекций. Большое внимание уделяется фондовой работе, в музее функционирует система CAMIS. Проводится оцифровка особо ценных исторических документов, таких как авторские свидетельства ученых, статьи и книги по теории управления, ранние работы Ю.И. Неймарка по D-разбиению. К празднованию золотого юбилея факультета электронный архив был дублирован на накопители и составил 0,5 Tflops.

2. Социальная роль музея

2.1. Учебная и воспитательная компоненты

Экспозиция музея активно используется преподавателями университета в учебном процессе. Доцент С.Г. Кузин строит свои лекции по истории развития средств вычислительной техники и архитектуре вычислительных систем так, чтобы музейные предметы наглядно подтверждали сказанное, а в рефераты по истории науки студенты могли бы включить фотосъемку экспонатов. Эти работы после защиты остаются в музейной библиотеке.

Экспозиция постоянно информирует своих посетителей о достижениях студенчества на международных конкурсах. Команда студентов нашего факультета в 2009 г. в Каире заняла 2-е место с проектом ViVa в финале международного конкурса Imagine CAP, проводимого компанией Microsoft, другая команда наших студентов заняла 1-е место в мире в конкурсе «Cluster Challenge» в Сизтле

в 2011 г. В экспозиции представлены результаты проектов ИТ- моделирования и анализа процессов различной природы, математическое моделирование, изучение динамики и оптимизации сложных дискретных и распределенных систем, систем массового обслуживания; разработка теоретических основ анализа многоэкстремальных моделей принятия решений, на лазерной плазме можно увидеть результаты современных исследований, проводимых на сложнейших установках [6].

2.2. Информационно-познавательная компонента

Территориальная открытость, подача материала в доступной форме с использованием компьютерных средств, уникальные собрания (русские счеты, логарифмические линейки, арифмометры, программируемые калькуляторы) заинтересовали студентов других факультетов, в том числе гуманитарных, музей стал популярен среди сотрудников и гостей университета.

Музей развивается. За пять учебных лет его посетило порядка 7 тыс. человек, проведено 418 экскурсий. В книге отзывов имеется нескольких десятков благодарственных записей от горожан и гостей города. А. Коновалов, участник одной из конференций, д.т.н., профессор, заведующий лабораторией Института машиноведения УрО РАН из Екатеринбурга, записал: «Музей истории вычислительной техники очень актуален, интересно и изящно оформлен». Ведущий инженер кафедры химии Т.И. Лиюгоньякая: «Случайно попав в ННГУ во второй корпус, я увидела музей факультета ВМК. Очень понравились собранные материалы, больше бы таких музеев!» Л.В. Сазонова: «Приехали на День открытых дверей с сыном – учеником 10 класса лицея г. Арзамаса. Наступила пора выбирать дисциплины, на которые делать упор в старших классах для поступления в престижный вуз. Получили полную информацию по всем интересующим вопросам и огромное впечатление от музея, от рассказа о поистине легендарных личностях! Я надеюсь, что мой сын без моего давления сделает правильный выбор». Р.С. Шкурко: «Большое спасибо за выставку! Она позволяет почувствовать дыхание времени и стремительность прогресса человеческого общества, позволяет задуматься о будущем».

Нужно ли говорить, как важно для сотрудников музея получить столь лестную оценку их скромного труда!

Музей принимает участие в семинарах кафедр и Ученого совета с сообщениями и юбилейными презентациями, участвует в издательской деятельности [7]. Вместе с сотрудниками библиотеки НИИ ПМК издан библиографический указатель Ю.И. Неймарка, на который получен отзыв от заведующей библиотекой физфака МГУ М.А. Знаменской. Отрывок из письма М.А. Знаменской: «Я давно, более 60 лет, работаю в области библиографии физики, но никогда не встречала библиографического указателя трудов ученого, выполненного не

только на таком высоком профессиональном уровне, но и сделанного с теплотой и любовью. Тексты, подбор фотографий, полиграфическое оформление – все заслуживает высочайшей оценки. Можно сказать, что это библиография замечательного ученого».

Заключение

Музей факультета ВМК является институтом социальной памяти, его миссия в том, чтобы показать, как развитие техники было связано с судьбами страны, нашего города, ученых и студентов университета.

Список используемых источников

1. Стронгин Р.Г. Культура и наука в современном мире / Р.Г. Стронгин. История и культура Нижегородского края. I Музейные научные чтения. 2000 г. Н. Новгород. ННГУ. 2003. С. 5-11.
2. Неймарк Ю.И. Сухой остаток. К истории в лицах научной школы А.А. Андропова. Н. Новгород, ННГУ. 2000.
3. Неймарк Ю.И. Вспоминая минувшие дни. К 45-летию ВМК ННГУ. Н. Новгород. ННГУ. 2008. С. 9-19.
4. Панкрашкина Н.Г. Образ техники в музее / Н.Г. Панкрашкина. История науки и техники: музейные исследования. Материалы научного семинара. 2-3 декабря 2008 г. – Н. Новгород, ННГУ. С. 103-105.
5. Эйнгорин М.Я. Как создавалась первая цифровая последовательная ЭВМ в СССР – машина ГИФТИ. – Н. Новгород, ННГУ. 2007.
6. Факультет вычислительной математики и кибернетики Нижегородского государственного университета им. Н.И. Лобачевского. – Н. Новгород, ННГУ. 2008.
7. Юрий Исаакович Неймарк. Библиографический указатель. К 45-летию факультета вычислительной математики и кибернетики. / Н.Г. Панкрашкина, Л.И. Фомина, Л.П. Корнюшина. – Н. Новгород, ННГУ. 2008.

Н.Г. Панкрашкина
Зав. музеем факультета ВМК
В.П. Савельев
Доцент кафедры ТУиДМ,
кандидат физ.-мат. наук,
декан факультета ВМК в 1990 - 2007 гг.

Доклад был опубликован в материалах Третьей международной конференции SORUCOM-2014, 13-17 октября, Казань

ПРИЛОЖЕНИЯ

Публикации по годам

1958

1. Кетков Ю.Л. Об одном способе вычисления полиномов на математических машинах. // Известия вузов. Радиофизика т.1, № 4, 1958

1959

1. Кетков Ю.Л. Схема сокращенного деления на ЦВМ последовательного действия. // Известия вузов. Радиофизика т. 2, № 2, 1959

1960

1. Кетков Ю.Л. Подпрограммы плавающей запятой. // Известия вузов. Радиофизика т. 3, № 1, 1960
2. Иванова И.М., Кетков Ю.Л., Ямпольская Т.С. О существовании кодов Баркера. // Известия вузов. Радиофизика т. 3, № 5, 1960

1961

1. Кетков Ю.Л. Прошивка координатной матрицы ЗУ на магнитных сердечниках, работающей по принципу многократного совпадения токов. // Известия вузов. Радиофизика т. 4, № 1, 1961

1964

1. Кетков Ю.Л. Система автоматизации плазовых расчетов (радиусография). – Ростов-на-Дону, НИТМ, 1964. – 98 с (под грифом «Секретно»)
2. Жеглова Н.В., Кетков Ю.Л. Система операций технологической машины. – Труды по вопросам применения ЭВМ в народном хозяйстве. – Горький, 1964
3. Антонова В.М., Баталова З.С., Кетков Ю.Л. Некоторые особенности программирования задач проектирования технологических карт. – Труды по вопросам применения ЭВМ в народном хозяйстве. – Горький, 1964

1966

1. Кетков Ю.Л. Об оптимальных методах кусочно-линейной аппроксимации. // Известия вузов. Радиофизика т. 9, № 6, 1966
2. Кетков Ю.Л. Автоматизация проектирования поверхностей корпусов судов. – В сб. «Автоматизация технологического проектирования при помощи ЭВМ». – М.: Машиностроение, 1966

3. Кетков Ю.Л. Об оптимальных методах нелинейной аппроксимации плоских кривых и системе автоматизации программирования для обработки геометрической информации. – Автореферат диссертации на соискание ученой степени кандидата физико-математических наук. Горький, 1966

1967

1. Кетков Ю.Л. О приближенных методах кусочно-линейной аппроксимации плоских кривых. – В сб. «Ученые записки. Прикладная математика и кибернетика. Труды НИИ ПМК и факультета ВМК ГГУ». – Горький, 1967
2. Кетков Ю.Л., Куракина И.М. О применении схем дифференциального спуска для минимизации функционалов и решения систем нелинейных уравнений. // Труды Всесоюзного симпозиума по прикладной математике и кибернетике. – Горький, 1967

1970

1. Кетков Ю.Л., Куракина И.М. Программа для решения систем нелинейных уравнений методом дифференциального спуска (ДЭС-1, ДЭС-2). // Аннотированный перечень алгоритмов и программ. – ВЦ МГУ, вып.1, 1970
2. Денисова С.Н., Кетков Ю.Л., Куркин Ю.С. Автоматизация расчетов уровней ожидаемой шумности в судовых помещениях. // Судостроение, №8, 1970
3. Кетков Ю.Л., Куракина И.М. Программа сноса точек на ЭВМ М-20. – В кн. А.Г.Угодчиков, М.И.Длугач, А.Е.Степанов «Решение краевых задач плоской теории упругости на цифровых и аналоговых машинах», М., 1970

1971

1. Белослудцева М.М., Бочаров В.В., Ильин Э.Н., Кетков Ю.Л., Куракина И.М., Максимов В.С. BASIC - система пакетной обработки задач на ЭВМ типа БЭСМ-4, М-220. Инструкция по эксплуатации (руководство). – Горький, 1971

1972

1. Белослудцева М.М., Бочаров В.В., Ильин Э.Н., Кетков Ю.Л., Куракина И.М., Максимов В.С., Шилова Ю.В. Использование алгоритмического разговорного языка BASIC на ЭВМ типа БЭСМ-4, М-220, М-222. // Труды Всесоюзного симпозиума «Теория языков и методы построения систем программирования». – Алушта, 1972

1973

1. Белослудцева М.М., Бочаров В.В., Ильин Э.Н., Кетков Ю.Л., Куракина И.М., Максимов В.С. BASIC - система пакетной обработки задач на ЭВМ типа БЭСМ-4, М-220, М-222. Учебное пособие. – Горький: изд. ГГУ, 1973. – 192 с
2. Кетков Ю.Л. Некоторые проблемы преподавания программирования. // Ученые записки ГГУ, 1973, вып. 186

1974

- 1 Кетков Ю.Л. Методические указания по изучению темы «Современные методы программирования» (методическое пособие). – М.: изд. ВЗИИТ, 1974. – 48 с
2. Гребенников А.И., Кетков Ю.Л. Программа «Профиль» аппроксимации профиля червячной фрезы дугами окружностей. // ГосФАП, рег. номер П000806, 1974

1975

1. Кетков Ю.Л., Куракина И.М. Стандартная программа ДЭС-3 минимизации функции многих переменных. // ГосФАП, рег. номер П001418, 1975
2. Кетков Ю.Л., Рябов А.Н. Инструкция по запуску и эксплуатации BASIC-системы на ЭВМ типа БЭСМ-4, м-220, М-222. // ГосФАП, рег. номер П001533, 1975

1976

1. Дутьшева Л.Я., Кетков Ю.Л., Кузин С.Г. Обучение программированию в СКП «Студент». – В сб. «Машинное обучение с помощью диалога». – Москва: МДНТП, 1976

1977

1. Кетков Ю.Л., Рябов А.Н. Программирование на БЭЙСИКЕ: Входной язык системы программирования БЭЙСИК-222. Учебное пособие. – Горький, 1977. – 73 с
2. Будеков О.А., Гололобов А.В., Кетков Ю.Л. Эвристические алгоритмы одномерного размещения связанных элементов. – В сб. «Вычислительная техника в машиностроении». – Минск, вып.2, 1977
3. Кетков Ю.Л., Куракина И.М., Тафорина Н.М. Программа определения оптимальной стратегии и цены игры. // Деп. статья, ВИНТИ, 3991-77 ДЕП

1978

1. Кетков Ю.Л. Программирование на БЭЙСИКЕ. –М.: Статистика, 1978. – 178 с
2. Кетков Ю.Л., Куракина И.М. Стандартная программа для решения матричных игр. // ГосФАП, рег. номер П003199, 1978
3. Кетков Ю.Л., Максимов В.С., Рябов А.Н. Входной язык программирования БЭЙСИК-222. // ГосФАП, рег. номер П003233, 1978
4. Кетков Ю.Л., Рябов А.Н. Эффективная система программирования BASIC - 222. //Программирование, 1978, №1
5. Кетков Ю.Л., Максимов В.С. Организация и работа с данными в специализированной САПР. – В сб. «Оптимизация и математическое обеспечение САПР». – Горький: изд. ГГУ, 1978

1979

1. Кетков Ю.Л. Системное программирование. Система команд ЕС ЭВМ. Анализ символьной информации. Учебно-методическое пособие. – М.: изд. ВЗИИТ, 1979. – 96 с

1980

1. Кетков Ю.Л. Программные средства для организации диалога с прикладной программой под управлением ДОС ЕС. – В сб «Оптимизация и математическое обеспечение САПР». – Горький: изд. ГГУ, 1980
2. Кетков Ю.Л. Макрокоманды анализа символьной информации. // ГосФАП, рег. номер П004370, 1980
3. Кетков Ю.Л., Максимов В.С., Тафорина Н.М. Форматные преобразования числовой информации, предназначенной для вывода на ЕС ЭВМ. // ГосФАП, рег. номер П004372, 1980
4. Кетков Ю.Л. Математическое обеспечение диалога с ПП под управлением ДОС ЕС. // Инф.листок МЦНТИ 105-80, Горький 1980

1981

1. Кетков Ю.Л., Куракина И.М. Математическое обеспечение графического диалога под управлением ДОС ЕС. – В сб «Оптимизация и математическое обеспечение САПР». – Горький: изд. ГГУ, 1981
2. Кетков Ю.Л., Максимов В.С. Естественное редактирование текстовых файлов в диалоговом режиме под управлением ДОС ЕС. – В сб. «Математическое моделирование и программное обеспечение САПР». – Горький: изд. ГГУ, 1981

-
3. Кетков Ю.Л. Пакет программ на ФОРТРАНЕ для решения целочисленных задач по исследованию операций на ЕС ЭВМ. // Инф. листок МЦНТИ 16-81 НТД, Горький
 4. Кетков Ю.Л. Редактор символьной информации в ДОС ЕС на базе дисплейной станции ЕС-7920. // Инф. листок МЦНТИ. 67-81 НТД, Горький

1982

1. Кетков Ю.Л., Максимов В.С., Рябов А.Н. Введение в системное программирование на языке АССЕМБЛЕРА ЕС ЭВМ. – М.: Наука, 1982. – 264 с

1983

1. Кетков Ю.Л., Куракина И.М. Программирование на алгоритмических языках БЕЙСИК и ФОРТРАН. Учебное пособие. – Горький: изд. ГГУ, 1983. – 104 с
2. Кетков Ю.Л. Пакет программ ДИАФОР для организации диалога под управлением ДОС ЕС. // ГосФАП, рег. номер П005892, 1983

1984

1. Кетков Ю.Л., Максимов В.С. Об одном этюде для программистов. // Программирование, 1984 № 3

1985

1. Гордион М.М., Кетков Ю.Л., Куракина И.М., Молоткова Т.Д., Тафорина Н.М. Программа воспроизведения ломаных линий и эквидистант на базе пакета ГРАФОР. // Деп. статья, ВИНИТИ, 4911-85 ДЕП

1986

1. Зверев В.И., Кетков Ю.Л., Максимов В.С. Алфавитно-цифровые дисплеи ЕС-7920 в диалоговых системах. – М.: Наука, 1986. – 240 с
2. Кетков Ю.Л., Куракина И.М. Программирование на алгоритмических языках БЕЙСИК и ФОРТРАН (изд. 2-е, дополн.). Учебное пособие. – Горький: изд. ГГУ, 1986. – 135 с
3. Кетков Ю.Л. Сердитое письмо - рецензия на книгу Р.Трейстора «Персональный компьютер фирмы ИВМ». // Микропрограммные средства и системы, 1986, №5

1987

1. Антипова Р.И., Белова М.Л., Гордион М.М., Ильин Э.Н., Кетков Ю.Л., Куракина И.М., Молоткова Т.Д., Тафорина Н.М. Инструментальная под-

система для вывода картографической информации на ЕС ЭВМ. – В сб. «Автоматизация обработки сложной графической информации». – Горький: изд. ГГУ, 1987

2. Кетков Ю.Л. Усовершенствование программы Schulmathematik. // Микропрограммные средства и системы, 1987, №4

1988

1. Кетков Ю.Л. Диалог на языке Бейсик для мини- и микро-ЭВМ. – М.: Наука, 1988. – 368 с
2. Кетков Ю.Л. Графические средства в алгоритмическом языке Бейсик. // Микропрограммные средства и системы, 1988

1990

1. Кетков Ю.Л. Размышления о БЕЙСИКЕ (три поколения популярного языка программирования). – В сб. «Математическое моделирование и оптимизация». – Горький: изд. ГГУ, 1990

1991

1. Кетков Ю.Л. Командно-файловый процессор Norton Commander. Версия 3.0. Спец. Практикум. – Н.Новгород: изд. ННГУ, 1991. – 42 с

1992

1. Кетков Ю.Л. *GW-, Turbo- и Quick-BASIC на IBM PC/XT и AT.* – М.: Финансы и статистика, 1992. – 240 с
2. Кетков Ю.Л. Толковый словарь языка программирования БЕЙСИК. – М.: Наука, 1992. – 320 с
3. Кетков Ю.Л. Norton Commander 3.0: Новые средства просмотра и редактирования файлов (Методическое пособие). – Н.Новгород: изд. НКЦП, 1992. – 86 с
4. Кетков Ю.Л. Создание инструментальных программных средств для разработки диалоговых систем САПР и АСНИ. – Диссертация на соискание ученой степени доктора технических наук в форме научного доклада, Н.Новгород, 1992

1994

1. Кетков Ю.Л. Двухоконный редактор текстовой информации для IBM PC. – В сб. «Математические модели и оптимальное управление», Н.Новгород: изд. ННГУ, 1994. – с.158-167

1996

1. Antipova R.I., Il'in E.N., Ketkov Yu.L. Time optimization in the preparation of display frames for visual control of cartographics materials in the KART-DOK system. // Pattern recognition and image analysis. – 1996, v. 6, № 1. – p. 212

1997

1. Кетков Ю.Л., Кетков А.Ю., Шапошников Д.Е. Школьная энциклопедия: Персональный компьютер. – М.: Большая Российская Энциклопедия, изд. Дом Дрофа, Москва, 1997. – 440 с
2. Кетков Ю.Л., Лебедев К.В., Петрова Е.В., Уварова Е.П. Технология визуального программирования в среде Delphi. Учебное пособие. – Н.Новгород: изд. ННГУ, 1997. – 138 с.

1998

1. Кетков Ю.Л., Кетков А.Ю., Шапошников Д.Е. Школьная энциклопедия: Персональный компьютер. – М.: Большая Российская Энциклопедия, изд. Дом Дрофа, Москва, 1998. – 440 с
2. Antipova R.I., Ketkov Yu.L. Optimization of a graphic file size and of the quality and time of imaging of mass-scale entities on layouts of towns. // Pattern recognition and image analysis. –1998. v.8. №3 p.285
3. Vasin Yu.G., Ketkov Yu.L., Kurakina I.M. KART-DOK: A system for the visualization of digital maps and the preparation of their edition // Pattern recognition and image analysis. –1998 v.8 №3. p. 302
4. Kazachkova M.S., Ketkov Yu.L., Lyutova N.V. Generation of large-size raster images. // Pattern recognition and image analysis. – 1998 v.8 №3 p. 318
5. Ketkov Yu.L. Transparent selection of the image fragments on a display. // Pattern recognition and image analysis. – 1998 т.8 №3 с. 319
6. Ketkov Yu.L., Kurakina I.M., Molotkova T.D. An approach to imaging of adjoining linear objects. // Pattern recognition and image analysis. –1998 v.8 №3 с.320
7. Ketkov Yu.L., Lyutova N.V. Creation of a character library based on the HP-GL standard. // Pattern recognition and image analysis. –1998 т.8 №3 с.321

1999

1. Кетков Ю.Л. Как преодолеть барьер Windows. // Межвуз.сб. Новые информационные технологии и развитие образования. Вып.286, ВГАВТ, Н. Новгород, 1999. – с. 29-42

2000

1. Кетков Ю.Л. Обработка временных интервалов и календарных дат на IBM PC. // Начертательная геометрия, инженерная и компьютерная графика. Международный межвузовский сб. трудов кафедр графических дисциплин, Н. Новгород, ННГАСУ, вып. 4, 2000. – с. 107-119
2. Кетков Ю.Л., Кетков А.Ю. Об одном подходе к подготовке профессиональных программистов. – В сб. ВГАВТ, Н.Новгород, 2000

2001

1. Кетков Ю.Л., Кетков А.Ю. Практика программирования: Бейсик, Си, Паскаль. Самоучитель. – Санкт-Петербург: БХВ-Петербург, 2001. – 480 с
2. Кетков Ю.Л., Казачкова М.С. Технология изготовления шаблонов печатных плат на IBM PC. Известия Академии инженерных наук РФ. Москва - Н. Новгород, 2001. с.111-125
3. Кетков Ю.Л., Кириянов С.К. Оптимизация времени отображения векторных графических изображений большого размера. Труды XI Международной конференции ГРАФИКОН-2001. Н. Новгород, 2001. – с.215-216

2002

1. Кетков Ю.Л., Кетков А.Ю. Практика программирования: Бейсик, Си, Паскаль. Самоучитель. – Санкт-Петербург: БХВ-Петербург, 2002. – 480 с
2. Кетков Ю.Л., Кетков А.Ю. Практика программирования: Visual Basic, C++ Builder, Delphi. Самоучитель. – Санкт-Петербург: БХВ-Петербург, 2002. – 464 с

2003

1. Кетков Ю.Л., Кузнецов А.И. Подготовка издательских оригиналов с помощью фотоплоттера КОРАЛ. – В юбилейном сб. научных статей «Математика и кибернетика 2003», изд. ННГУ, 2003, с.162-164

2004

1. Кетков Ю.Л., Кетков А.Ю., Шульц М.М. MatLab 6х: программирование численных методов. – Санкт-Петербург: БХВ-Петербург, 2004. – 672 с

2005

1. Кетков Ю.Л., Кетков А.Ю., Шульц М.М. MatLab 7: программирование, численные методы. – Санкт-Петербург: БХВ-Петербург, 2005. – 752 с
2. Кетков Ю.Л., Медвецкая М.А. Электронное пособие по изучению алгоритмов машинной графики. // VIII Всероссийская конференция «Методы

и средства обработки сложной графической информации», сент. 2005 г. Нижний Новгород

3. Кетков Ю.Л., Кузнецов А.И. Обыкновенные дифференциальные уравнения: MatLab versus MathCAD. // Математика в высшем образовании. – Н,Новгород, изд. ННГУ, №3. 2005. с.27-52
4. Y. Ketkov, M. Tyulin. Real-time visualization of large vector HP-GL based maps. – Труды 15-й Международной конференции по компьютерной графике и ее приложениям GraphiCon'2005, Новосибирск, 2005 г, pp. 316-317
5. Кетков Ю.Л., Кузнецов А.И. Создание графических примитивов в формате HP-GL. – там же, с. 378-381

2006

1. Кетков Ю.Л. Введение в языки программирования С и С++. Интернет-университет информационных технологий - ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2006 г., 344 стр

2008

1. Кетков Ю.Л., Матвеев З.А. Методы повышения эффективности систем воспроизведения картографических документов. // Вестник ННГУ. - изд. ННГУ, № 2, 2008, с. 138-146

2010

1. Кетков Ю.Л., Матвеев З.А. Анализ алгоритмов оптимизации времени отображения электронных карт в формате HP-GL // Графикон-2010: труды 20-й международной конференции по компьютерной графике и зрению / Санкт-Петербургский гос. унт-т инф. технологий, механики и оптики. – СПб., 2010. – С. 246-252

2011

1. Кетков Ю.Л., Кетков А.Ю. Свободное программное обеспечение: Free Pascal для студентов и школьников. – Санкт-Петербург: БХВ-Петербург, 2011. – 384 с
2. Кетков Ю.Л. Школа программирования ИПМ им. акад. М.В.Келдыша // Труды SORUCOM-2011. Вторая Международная конференция «Развитие вычислительной техники и ее программного обеспечения в России и странах бывшего СССР. – Великий Новгород, 2011. – с.137 – 142
3. Кетков Ю.Л., Матвеев З.А. Автоматическое создание кластерной модели графического образа электронных карт в формате HP-GL. // Приволжский научный журнал, 2011, №1, с.37-41

4. Кетков Ю.Л., Матвеев З.А. Разработка и исследование алгоритмов визуализации цифровых карт в векторном формате. // Приволжский научный журнал, 2011, №2, с.45-49

2012

1. Кетков Ю.Л., Кузнецов А.И., Леденцов Д.К., Подогова С.Д. Совершенствование функциональных свойств визуализатора электронных карт. // Вестник ННГУ. - изд. ННГУ, №5, 2012. С.367-375.
2. Кетков Ю.Л. Школа программирования ИПМ им. акад. М.В.Келдыша [Электронный ресурс]. Статья размещена на сайте ИПМ РАН (www.keldysh.ru/events/ketkov) 22.12.2012
3. Кетков Ю.Л., Кузнецов А.И. Использование TTF-шрифтов в системах визуализации электронных карт // Приволжский научный журнал. – Н.Новгород, 2012. – №2. – С.115-119

2013

1. Кетков Ю.Л. Они были первыми. // Раздел в сб. Мотова М.И., Шалфеев В.Д. А.А.Андронов и зарождение кибернетики в Нижегородском университете. – Н.Новгород, изд-во ННГУ, 2013. С.47-62

Научно-технические отчеты по НИР и ОКР

№ п/п	Наименование отчета
1964 (ГИФТИ)	
1	Интерпретирующая система радиосографического метода (ИС-РГМ)
2	Универсальная ИС-РГМ (УИС-РГМ)
3	Развертка листов судовой поверхности
1967	
4	Вычислительный комплекс университета и его математическое обеспечение (отчет ГК НТ СССР)
1968 (НИИ ПМК)	
5	Алгоритмы и стандартные программы для решения детерминированных задач нелинейного программирования (отчет ГК НТ)
1969	
6	Стандартные программы ИС-2 для ЭВМ типа М-20 (4 отчета по х/д 6910)
7	Стандартная программа редактирования и печати программ на языке АЛГОЛ-60
1971	
8	Автоматизация проектных и конструкторских работ в судостроении (х/д 7129 с предприятием п/я А-1932)
9	Х/д 6812-А с предприятием п/я В-2715 (тема ОБ-29 «Чертеж» по постановлению ВПК при СМ СССР)

1972	
10	Автоматизация проектных и конструкторских работ в судостроении (х/д 7129 с предприятием п/я А-1932)
11	Х/д 6812-А с предприятием п/я В-2715 (тема ОБ-29 «Чертеж» по постановлению ВПК при СМ СССР)
1973	
12	Система МО «Дубна» для ЭВМ типа М-20
13	Т-413 (отчет по теме «Чертеж», х/д 6812)
1974	
14	Т-429 (отчет по теме «Чертеж»)
15	Т-437 (отчет по теме «Чертеж»)
16	Т-438 (отчет по теме «Чертеж»)
1975	
17	Бейсик-система пакетной обработки задач на ЭВМ М-222 (в 4-х частях)
18	Инструкция по эксплуатации транслятора ТА-2М (х/д 7601)
19	Интерпретирующая система ИС-22 и библиотека стандартных программ (х/д 7601)
20	Т-454 (отчет по теме «Чертеж»)
21	Т-455 (отчет по теме «Чертеж»)
22	Т-456 (отчет по теме «Чертеж»)
23	Т-459 (отчет по теме «Чертеж»)
24	Т-460 (отчет по теме «Чертеж»)

1976	
25	Т-499 (отчет по теме «Чертеж»)
26	Т-500 (отчет по теме «Чертеж»)
27	Т-501 (отчет по теме «Чертеж»)
28	Т-507 (отчет по теме «Чертеж»)
29	Пакет научно-исследовательских программ на Фортране для ЕС-1020
30	2 отчета по х/д 03.7601 (КБЭ, г. Сарапул)
31	Система пакетной обработки BASIC-222 (в 4-х частях)
1977	
32	Т-515 (отчет по теме «Чертеж», х/д 03.7627)
33	Спецприложение к отчету Т-515 (тема «Чертеж»)
34	Т-516 (отчет по теме «Чертеж»)
35	Отчет по х/д 7632. Анализ системы проектирования ОКБ
36	Отчет по х/д 7657. Требования и правила оформления программной документации в системе САПР ПКО «Гидромаш»
1978	
37	Отчет по х/д 7657. Справочное руководство по эксплуатации комплекса графических программ ГРАФОР на ЭВМ типа ЕС-1020 и М-4030
38	Отчет по х/д 7668. Система коллективного пользования на базе ЕС-1020 (1022). Описание входного языка.
39	Отчет по х/д 7713. Требования и правила для оформления программной документации для САПР предприятия п/я Р-6328

40	Комплекс программ подготовки управляющей информации на перфоленте для автоматического координатографа КПА-1200
41	Разработка эмулятора М-6000 на ЕС ЭВМ
42	Разработка информационно-справочной системы для ЭВМ типа ЕС-1020
1979	
43	НИР «КАСИПР». Разработка пакета программ воспроизведения условных знаков топографических карт масштабов 1:25000, 1:50000, 1:100000
44	НИР «МОДЕЛЬ». Математическое обеспечение ЕС ЭВМ для вывода графической информации на графопостроители типа ЕС-7251 и КПА-1200. – 59 с.
1980	
45	Перевод программ подсистемы «Шимми» САПР 1-й очереди из пакетного режима в диалоговый (х/д 7913)
46	Пакет программ для организации диалога под управлением ДЭС-ЕС
47	Пакет программ на Фортране для решения целочисленных задач по исследованию операций
48	Система коллективного пользования на базе ЕС ЭВМ. Структура и выполнение рабочей программы
49	НИР «КАСИПР». Пакет унифицированных программ воспроизведения точечных знаков и подписей для топографических карт масштабов 1:25000, 1:50000, 1:100000. – 155 с.
1981	
50	Редактор символьной информации на базе дисплейной станции ЕС-7920

51	НИР «КАСИПР». Пакет программ для воспроизведения линейных и площадных знаков для топографических карт масштабов 1:25000, 1:50000, 1:100000
52	НИР «КАСИПР». Разработка комплексной программы воспроизведения точечных, линейных и площадных знаков топографических карт масштабов 1:25 000, 1:50 000, 1:100 000
53	Базовые программы для обмена с дисплейной станцией ЕС-7920 под управлением ДОС-ЕС (руководство программиста)
54	Базовые программы для обмена с дисплейной станцией ЕС-7920 под управлением ДОС-ЕС (руководство системного программиста)
55	Работа с локальной дисплейной станцией ЕС-7920 под управлением ДОС-ЕС (руководство оператора)
56	Комплекс программ ЕС-ЭВМ для организации связи с АРМ-М
1982	
57	Развитие диалоговых средств общения с прикладной программой под управлением ДОС-ЕС и ОС-ЕС
58	Диалоговый редактор символьной информации на базе дисплейной станции ЕС-7920 (Версия 2)
59	Подсистема вывода картографической информации на базе ЕС ЭВМ. Технический проект. – 248 с
60	Разработка технических и программных средств обеспечения машинной графики и диалога на базе ЕС-1022 (х/д 03.8225 – тема ДИАГРАФ с предприятием п/я Г-4453)
1983	
61	Инструкция для начинающих по работе в системе коллективного доступа PRIMUS
62	Технические и программные средства машинной графики в составе ДОС ЕС (х/д 8225 с ПМЗ «ВОСХОД», тема ДИАГРАФ)

63	Технические и программные средства машинной графики в составе ОС ЕС (х/д 8230 с Павловским механическим заводом, тема ГРАФИК)
1984	
64	Многопультный дисплейный редактор символьной информации для ДОС-ЕС. Общее описание
65	« → » – « . Руководство программиста
66	« → » – « . Генерация системы
67	« → » – « . Руководство оператора
68	« → » – « . Описание программ
69	« → » – « . Тексты программ
70	АКВАРЕЛЬ. КЦТР.00131. ВЫВОД. Описание применения. – 65 с.
71	« → » – « . Спецификация. – 4 с.
72	« → » – « . Руководство системного программиста. – 65 с.
73	АКВАРЕЛЬ. КЦТР.00177. Подсистема ВЫВОД. Пакет программ ДИАФОР ЕС для организации диалога на Фортране между пользователем и прикладной программой. Тексты программ. – 91 с.
74	« → » – « . Описание применения. – 21 с.
75	« → » – « . Руководство оператора. – 10 с.
76	« → » – « . Руководство программиста. – 59 с.
77	« → » – « . Руководство системного программиста. – 23 с.
78	АКВАРЕЛЬ. КЦТР.00180. Подсистема вывода. Комплекс программ воспроизведения дискретных знаков. Описание программы. – 15 с.

79	« →» – « . Руководство программиста. – 20 с.
80	« →» – « . Руководство системного программиста. – 16 с.
81	« →» – « . Тексты программ. – 172 с.
82	АКВАРЕЛЬ. КЦТР.00181. Подсистема вывода. Комплекс программ воспроизведения линейных знаков. Описание программы. – 44 с.
83	« →» – « . Руководство программиста. – 22 с.
84	« →» – « . Руководство системного программиста. – 16 с.
85	« →» – « . Тексты программ. – 51 с.
86	АКВАРЕЛЬ. КЦТР.00182. Подсистема вывода. Комплекс программ воспроизведения площадных знаков. Описание программы. – 13 с.
87	« →» – « . Руководство программиста. – 20 с.
88	« →» – « . Тексты программ. – 19 с.
89	АКВАРЕЛЬ. КЦТР.00183. Подсистема вывода. Комплекс программ воспроизведения надписей. Описание программы. – 15 с.
90	« →» – « . Руководство программиста. – 24 с.
91	« →» – « . Тексты программ. – 6 с.
92	АКВАРЕЛЬ. КЦТР.00184. Подсистема вывода. Комплекс программ расчета и воспроизведения рамки. Описание программы. – 21 с.
93	« →» – « . Руководство программиста. – 18 с.
94	« →» – « . Тексты программ. – 25 с.
95	Отчет по х/д 8334 (Спецтема)

96	Отчет по х/д 03.8435 с предприятием п/я М-5068 (тема «Луна»)
1985	
97	АСНИ АКВАРЕЛЬ. Подсистема ВЫВОД. Спецификация
98	« →» – « . Руководство программиста
99	« →» – « . Руководство системного программиста
100	« →» – « . Описание применения
101	« →» – « . Программа и методика испытаний
102	АСНИ АКВАРЕЛЬ. ПП ДИАФОР ЕС. Спецификация
103	« →» – « . Тексты программ
104	« →» – « . Описание применения
105	« →» – « . Руководство программиста
106	« →» – « . Руководство системного программиста
107	« →» – « . Руководство оператора
108	АСНИ АКВАРЕЛЬ. Комплекс программ воспроизведения дискретных знаков. Спецификация
109	« →» – « . Тексты программ
110	« →» – « . Описание применения
111	« →» – « . Руководство программиста
112	« →» – « . Руководство системного программиста
113	АСНИ АКВАРЕЛЬ. Комплекс программ воспроизведения линейных знаков. Спецификация
114	« →» – « . Тексты программ

115	« →» – « . Описание применения
116	« →» – « . Руководство программиста
117	« →» – « . Руководство системного программиста
118	АСНИ АКВАРЕЛЬ. Комплекс программ воспроизведения площадных знаков. Спецификация
119	« →» – « . Тексты программ
120	« →» – « . Описание применения
121	« →» – « . Руководство программиста
122	« →» – « . Руководство системного программиста
123	АСНИ АКВАРЕЛЬ. Комплекс программ воспроизведения подписей. Спецификация
124	« →» – « . Тексты программ
125	« →» – « . Описание применения
126	« →» – « . Руководство программиста
127	« →» – « . Руководство системного программиста
1986	
128	Т-732 (Тема «Луна»)
129	Т-742 (Тема «Типаж»)
130	Т-751 (Тема «Типаж»)
131	Подсистема ВЫВОД-ТИПАЖ. – 105 с.
1987	

132	АСОИМК – автоматизированная система обработки информации морских карт. Подсистема воспроизведения картографических документов ВЫВОД-АСОИМК. КЦТР.00216. Описание применения. – 38 с.
133	АСОИМК. Подсистема формирования библиотеки описаний дискретных знаков и символов шрифтов (БИБ КПА). КЦТР.00217. Описание применения. – 10 с.
134	« →» – « . Описание программы. – 61 с.
135	« →» – « . Описание языка. – 27 с.
136	« →» – « . Руководство системного программиста. – 164 с.
137	« →» – « . Тексты программ. – 37 с.
138	Разработка специализированного программного обеспечения для ЭВМ типа СМ-3(4), Электроника-60 для реализации вычислений с повышенной точностью
139	Программа-календарь для ЕС ЭВМ
1988	
140	Подсистема ВЫВОД-АСОИМК. КЦТР.00216. Описание программы
141	« →» – « . Тексты программ. – 186 с.
	« →» – « . Руководство оператора. – 17 с.
142	АКС ЛЕС-1: автоматизированная система обработки планшетов с лесоустроительной информацией. Подсистема ВЫВОД. Спецификация
143	« →» – « . Описание применения
144	« →» – « . Тексты программ
145	« →» – « . Описание программы. – 183 с.

146	« →» – « . Руководство оператора
147	Отчет по х/д 8613 (Тема «Локомотив»)
1989	
148	Т-811 (Тема «Локомотив»)
149	АСНИ АКВАРЕЛЬ-2. Подсистема ВЫВОД. Спецификация
150	« →» – « . Программа и методика испытаний
151	« →» – « . Описание применения
152	« →» – « . Тексты программ
153	« →» – « . Руководство оператора
154	Т-813 (Тема «Карта»)
1991	
155	АКВАРЕЛЬ-3. Инструментальный комплекс ZNAK-PC для создания библиотек и отображения картографических шрифтов и условных знаков на персональных ЭВМ типа IBM PC
156	АКВАРЕЛЬ-3. Методы и алгоритмы символизации условных знаков и шрифтов картографических документов
1992	
157	АРМ РАСТР. Автоматизированная система создания цифровых топографических карт. КЦТР.00313. Описание применения. – 19 с.
158	« →» – « . Руководство оператора. – 13 с.
1993	
159	СУАБДЦК. Система управления архивом баз данных цифровых топографических карт масштабов 1:10 000 – 1:1 000 000. КЦТР.00319. ВЫВОД. Пояснительная записка. – 21 с.

160	АРМ РАСТР-2. Автоматизированная система создания цифровых топографических карт масштабов 1:10 000 – 1:100 000. КЦТР.00321. Эскизно-технический проект. ВЫВОД. Пояснительная записка. – 21 с.
1994	
161	СУАБДЦК. Система управления архивом баз данных цифровых топографических карт масштабов 1:10 000 – 1:1 000 000. КЦТР.00319. ВЫВОД. Описание программы. – 121 с.
162	« →» – « . Руководство оператора. – 10 с.
163	« →» – « . Тексты программ. – 320 с.
164	« →» – « . Описание применения. Рабочий проект. – 18 с.
165	« →» – « . Тексты программ. Рабочий проект. – 150 с.
166	« →» – « . Описание программы. Рабочий проект. – 134 с.
167	АРМ КАРТ-ДОК. Разработка опытного образца автоматизированного рабочего места и технологии изготовления издательских оригиналов топографических карт масштабов 1:25 000 – 1: 1 000 000 000 по цифровым картам тех же масштабов. КЦТР.00320. Эскизно-технический проект. Пояснительная записка. – 98 с.
168	« →» – « . ВЫВОД. Пояснительная записка. – 154 с.
169	« →» – « . ВЫВОД. Руководство оператора. – 19 с.
170	« →» – « . ЗНАК. Руководство оператора. – 20 с.
171	« →» – « . Описание орг. структуры. – 10 с.
172	« →» – « . Описание применения. – 15 с.
173	« →» – « . Описание технологического процесса. – 21 с.
174	« →» – « . Руководство пользователя. – 31 с.
175	« →» – « . Руководство системного программиста. – 18 с.

176	« →» – « . Технологическая инструкция. – 20 с.
177	АРМ РАСТР-2. Автоматизированная система создания цифровых топографических карт масштабов 1:10 000 – 1:100 000. КЦТР.00321. ВЫВОД. Описание программы. – 185 с.
178	« →» – « . Руководство оператора. – 19 с.
179	« →» – « . Руководство пользователя. – 20 с.
1995	
180	СУАБДЦК. Система управления архивом баз данных цифровых топографических карт масштабов 1:10 000 – 1:1 000 000. КЦТР.00319. ВЫВОД. Описание программы. – 121 с.
181	« →» – « . Руководство оператора. – 12 с.
182	КЦТР.00320.ВЫВОД. Описание программы. – 202 с.
183	« →» – « . ВЫВОД. Руководство системного программиста. – 19 с.
184	« →». Описание применения. – 50 с.
185	« →». Описание технологического процесса. – 23 с.
186	« →». Руководство пользователя. – 33 с.
187	« →». Руководство системного программиста. – 19 с.
188	« →». Технологическая инструкция. – 22 с.
1996	
189	АРМ КАРТ-ДОК. Разработка опытного образца автоматизированного рабочего места и технологии изготовления издательских оригиналов топографических карт масштабов 1:25 000 – 1: 1 000 000 000 по цифровым картам тех же масштабов. КЦТР.00320. Описание технологического процесса. – 21 с.
190	« →». Технологическая инструкция. – 26 с.

191	« →». Руководство системного программиста. – 21 с.
192	« →». Руководство пользователя. – 33 с.
193	КЦТР.00320. ЗНАК. Руководство системного программиста. – 22 с.
194	АРМ РАСТР-2. Автоматизированная система создания цифровых топографических карт масштабов 1:10 000 – 1:100 000. КЦТР.00321. ВЫВОД. Описание применения. – 19 с.
195	« →» – « . Руководство оператора. – 19 с.
1997	
196	АРМ КАРТ-ДОК. Описание программы. – 16 с.
197	АРМ КАРТ-ДОК.ВЕКТОР_РАСТР. Описание программы. – 24 с.
198	АРМ КАРТ-ДОК. ВЫВОД. Руководство оператора. – 25 с.
199	АРМ КАРТ-ДОК. ЗНАК. Руководство оператора. – 17 с.
200	АРМ РАСТР-2. Автоматизированная система создания цифровых топографических карт масштабов 1:10 000 – 1:100 000. КЦТР.00321. Руководство системного программиста. – 19 с.
201	ЭКРАН. Система символизации картографического изображения на экранах коллективного и индивидуального пользования. КЦТР.00329. Эскизно-технический проект. Пояснительная записка. – 44 с.
201	« →» – « . ЗНАК. Пояснительная записка. – 26 с.
202	« →» – « . ВЕКТОР-ЭКРАН. Пояснительная записка. – 26 с.
1998	
203	ГИС «ТЕРРА». Подсистема нормативного отображения цифровых топографических карт и планов городов на экране дисплея. Руководство пользователя. – 22 с.

204	ГИС «ТЕРРА». Получение векторных и растровых графических файлов издательских оригиналов топографических карт и планов городов. Руководство пользователя. – 16 с.
205	АРМ КАРТ-ДОК. РАСТР. Описание программы. – 25 с.
206	ЭКРАН. ВЕКТОР-РАСТР. Описание программы. – 187 с.
207	« →» – « . Руководство оператора. – 47 с.
208	« →» – « . Руководство пользователя. – 13 с.
209	ЭКРАН. ЗНАК. Описание программы. – 41 с.
210	« →» – « . Руководство оператора. – 17 с.
211	« →» – « . Руководство пользователя. – 11 с.
1999	
212	ЭКРАН. Система символизации картографического изображения на экранах коллективного и индивидуального пользования. КЦТР.00329. Описание программы. – 201 с.
213	« →». ВЕКТОР-РАСТР. Руководство пользователя. – 12 с.
214	« →». ВЕКТОР-РАСТР. Руководство оператора. – 50 с.
215	« →». ЗНАК. Руководство пользователя. – 14 с.
216	« →». ЗНАК. Описание программы. – 41 с.
217	« →». ЗНАК. Руководство оператора. – 21 с.

Не включены отчёты, сделанные после 2000 г. (*примеч. ред.*)

Содержание

Юлий Лзаревич Кетков

Предисловие6
Р.Г. Стронгин

Из когорты легендарных личностей.....9
В.П. Гергель

Краткая биография Юлия Лазаревича Кеткова.....13

Мемуары. Избранные статьи

Пятьдесят лет с компьютером19

Об одном этюде для программистов.....64

Об одном способе вычисления полиномов на математических
машинах72

Схема сокращённого деления для цифровых вычислительных машин
последовательного действия.....79

О существовании кодов Баркера81

Прошивка координатной матрицы запоминающего устройства на
магнитных сердечниках.....84

Об оптимальных методах нелинейной аппроксимации плоских кривых и
системе автоматизации программирования для обработки геометрической
информации. Автореферат88

Об оптимальных методах кусочно-линейной аппроксимации.....97

Автоматизация проектирования поверхностей корпусов судов
при помощи ЭВМ.....105

О приближённых методах кусочно-линейной аппроксимации плоских
кривых124

Создание инструментальных программных средств для разработки
диалоговых систем САПР и АСНИ135

Она была первой160

Машина ГИФТИ	163
Школа программирования ИПМ им. акад. М.В. Келдыша	170
Как преодолеть барьер Windows?.....	191
Об одном подходе к подготовке профессиональных программистов.....	199
О некоторых пионерских работах на первых ЭВМ.....	203
Разработка диалоговых систем программирования в Нижегородском университете.....	217
Информатизация в средней школе на базе алгоритмического языка БЕЙСИК.....	226
Запросы на установку транслятора Бейсик.....	231
Фотографии.....	235
Воспоминания о Ю.Л. Кеткове	
Близкое и далекое.....	263
<i>И.В. Кеткова</i>	
Мой друг Юля Кетков.....	276
<i>М.М. Шульц</i>	
Слово о друге и коллеге.....	278
<i>С.И. Ротков</i>	
Человек с большим сердцем.....	282
<i>Т.Д. Молоткова</i>	
Учитель с большой буквы.....	283
<i>О.А. Козлов</i>	
Тот самый Кетков!.....	286
<i>А.Н. Рябов</i>	
Воспоминания об отце	291
<i>А.Ю. Кетков</i>	
Небожитель.....	294
<i>З.А. Матвеев</i>	

Ю.Л. Кетков – пионер в Нижегородском программировании300
А.И. Кузнецов

Музей факультета ВМК ННГУ в образовательном процессе307
Н.Г. Панкрашкіна, В.П. Савельев

Приложения

Публикации по годам315

Научно-технические отчеты по НИР и ОКР326

