

# ИНФОРМАТИКА И ОБРАЗОВАНИЕ

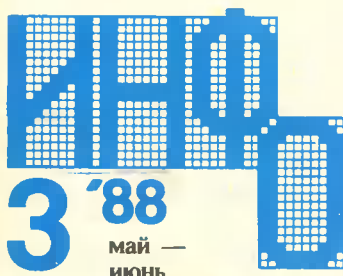
3 1988





**В этом электронном устройстве 42 микросхемы. А если бы его пришлось собирать из отдельных радиодеталей, одних транзисторов понадобилось бы около 150 000 штук, не говоря уже о резисторах и конденсаторах.**

**На первой странице обложки: дисплейный класс на базе ДВК-3 с черно-белой графикой, оборудованный системой РВС МИТХТ (статью о ней читайте в этом номере журнала).**



# ИНФОРМАТИКА И ОБРАЗОВАНИЕ

## Содержание

- Северный А. 40 лет новой эры 3  
Фурсенко А. Компьютерная технология обучения в средних специаль-  
ных учебных заведениях 5

## Методика обучения

- Гейн А., Житомирский В., Линецкий Е., Сапир М., Шолохович В. 11  
Программно-методический комплекс по курсу школьной информатики  
Утлинский Е., Степанов А., Ермакова В. Дидактический материал для  
текущего контроля знаний 25  
Штильман З. Практикум: программирование на основе алгоритмиче-  
ского языка 35  
Каймин В., Федюшин Д., Щеголев А. Методика преподавания инфор-  
матики в компьютерной физматшколе МИЭМ 43

## КВТ

- Архангельский А. Мир ЭВМ 48  
Дименштейн Р. «Прозаические редакторы» на ЭВМ 55  
Анисимов В., Соловьев Л. Учебный пакет прикладных программ  
«МОТОР» 63  
Савченко А., Филиппов А., Колосов К., Полетаев В., Федоров П., 74  
Пасхин А. Разработка обучающих систем  
Корнюшко В., Авдеев В., Фролов Г., Жедь А. Распределенная 78  
вычислительная система МИТХТ  
Гиглавый А. Новый персональный компьютер «Архимед» 80  
Обслуживание и ремонт вычислительной техники 81

## Внеклассная работа

- Матиясевич Ю. Задачи первых ленинградских олимпиад по информа- 85  
тике

## Педагогический опыт

- Говорунов Г., Гушина М., Ильченко И., Портяненко Н. Среднее 88  
медицинское училище и компьютер  
Иванова О. Опыт обучения работе на клавиатуре персональных ЭВМ 94  
Бузин А. «Агат» в нашей школе 98  
Звавич Л. Информатика на уроках математики 101

## Молодежная инициатива

103

## Точка зрения

- Родионов А., Родионова Н. Диалог в учебных программах 108  
Викентьев Л., Козлов О. Наше мнение 109

## ЭВМ в народном хозяйстве

- Фомин В. Компьютеры в музее 111  
Музей в компьютере 113

- После конкурса 114

## Нам пишут

- Компьютеризация в школе и развитие самостоятельного мышления у учащихся 116  
Об алгоритмической нотации 117

## Информация

- «Полиграфбуммаш — 87» глазами специалистов 120  
СССР — Болгария: сотрудничество в области образования 123  
Семинар в Воронеже 124

## Книги

- Дети, компьютеры и плодотворные идеи 125

- Веселый урок 128

Обложка Э. Бажилина

В оформлении номера принимали участие Э. Бажилин, Р. Бумагин, О. Гаспарянц, А. Лобашинский, А. Пономарев, С. Расторгуев.

Главный редактор  
академик

**В. А. МЕЛЬНИКОВ**

Редакционная  
коллегия

**И. М. БОБКО**

**Б. М. ГЕРАСИМОВ**

**Г. В. ГОДЖЕЛЛО**

**А. В. ДЕНИСЕНКО**

**А. П. ЕРШОВ**

**С. А. ЖДАНОВ**

**Б. В. ЛОМОВ**

**Ю. В. ЛУИЗО**

(зам. главного

редактора)

**Н. Г. МЕЛЬДИАНОВ**

**И. С. ОРЕШКОВ**

**О. К. ПАВЛОВА**

**А. Ю. УВАРОВ**

**А. И. ФУРСЕНКО**

**В. О. ХОРОШИЛОВ**

Редактор отдела **Ф. Даниловский**  
Редактор отдела **К. Шеховцев**  
Научный редактор **Т. Драгньш**  
Заведующая редакцией **Н. Игнатова**  
Художественный редактор **Л. Розанова**  
Корректор **Л. Чичулина**

Издательство «Педагогика» Академии педагогических наук СССР и Государственного комитета СССР по делам издательств, полиграфии и книжной торговли

Почтовый адрес: 107005, Москва, Лефортовский пер., 8.  
Телефон редакции: 249-97-77

Сдано в набор 23.03.88. Подписано в печать 26.04.88. А 05683.  
Формат 70×100/16. Печать офсетная. Усл. печ. л. 10,40. Уч.-изд. л. 13,26.  
Усл. кр.-отт. 42,88. Тираж 95 990 экз. Заказ 713. Цена 60 коп.

Ордена Трудового Красного Знамени Чеховский полиграфический комбинат  
ВО «Союзполиграфпром» Государственного комитета СССР по делам издательств,  
полиграфии и книжной торговли.  
142300, г. Чехов Московской области.

© Издательство «Педагогика», «Информатика и образование», 1988

А. СЕВЕРНЫЙ

## 40 лет новой эры

«Вчера фирма «Белл Телефон Лейбороториз» (Вест-стрит, 463) впервые продемонстрировала изобретенный ею прибор под названием «транзистор», который в некоторых случаях можно использовать в области радиотехники вместо электронных ламп.

Прибор был продемонстрирован в схеме радиоприемника, не содержащей обычных ламп. Было также показано его применение в телефонной системе и в телевизионном устройстве, управляемом с помощью приемника, расположенного на другом этаже. В каждом из этих случаев транзистор использовался в качестве усилителя, хотя фирма заявляет, что он может использоваться и в качестве генератора, способного создавать и передавать радиоволны.

Транзистор, имеющий форму металлического цилиндра длиной около 13 мм, не содержит полости, из которой откачан воздух, сетки, анода или стеклянного корпуса, предохраняющего от попадания в прибор воздуха. Он начинает работу мгновенно, без задержки на разогрев, так как в отличие от радиолампы в нем нет накала.

Рабочие элементы прибора состоят всего лишь из двух тонких проволочек, подходящих к кусочку твердого полупроводникового материала величиной с булавочную головку, припаянному к металлическому основанию. Вещество, помещенное на металличе-

ском основании, усиливает ток, подводимый к нему по одной проволочке, а другая проволочка отводит усиленный ток».

Это сообщение было помещено в газете «Нью-Йорк таймс» 1 июля 1948 г. Следует отметить, что первые полупроводниковые генератор и усилитель создал советский инженер О. В. Лосев. В 1932 г. он сконструировал кристаллин — радиоприемник, усилительным элементом которого был полупроводниковый диод.

Исследования в области физики и химии полупроводников проводились с конца 20-х гг. в разных странах. Во время второй мировой войны интерес к ним значительно возрос. Использование полупроводниковых датчиков для радиолокационных станций представлялось настолько важным в военном отношении, что правительство США выделило специальные средства на проведение программы по исследованию свойств кремния и германия. Однако с окончанием войны энтузиазм угас, и большинство исследователей вернулись к имевшим большее практическое применение разработкам электровакуумных приборов.

Однако «Белл Лейбороториз» свои работы продолжала. Под руководством В. Шокли и С. О. Моргана была создана группа по исследованиям в области физики твердого тела. Активная работа привела к тому, что в

декабре 1947 г. Д. Бардин и У. Браттейн создали полупроводниковый усилитель с коэффициентом усиления по напряжению порядка 100; усилитель работал вплоть до верхней границы диапазона звуковых частот.

Шокли не принимал непосредственного участия в открытии точечного транзистора (патент был выдан Бардину и Браттейну), но ему принадлежал основной вклад в теорию, приводящую к открытию транзистора. В 1956 г. всем троем была присуждена высшая научная награда — Нобелевская премия по физике за исследования полупроводников и открытие транзисторного эффекта.

4 В чем же была революционность происходящего? Лишь в замене хрупкой, сложной по конструкции электронной лампы не менее хрупким и сложным в изготовлении, хотя и более компактным, транзистором? Отнюдь нет! Конечно, если рассматривать тот, первый экземпляр, то особых преимуществ мы не увидим, разве что мгновенное начало работы и более низкое напряжение питания. Однако что представляет собой транзистор с точки зрения физиков? Это не проволочки и металлическое основание, это несколько (в классическом варианте — три) областей в толще кристалла кремния или германия, имеющих: одни — избыток свободных электронов, другие — недостаток. На границах разноименных областей (которые называют *n*- и *p*-переходами) под действием электрического поля происходит усиление протекающего через них тока. Заслуга Шокли, Бардина и Браттейна в том, что они смогли прикоснуться к пониманию процессов переноса зарядов в таких областях и на границах между ними. Если рассматривать зависимость протекающего тока от напряженности электрического поля, то оказывается, что совокупность областей *n*- и *p*-типов может вести себя не только как диод или усилитель, но и как резистор, конденсатор, а в более сложных случаях и как индуктивность.

Технологи нашли способ изготовления на одной полупроводниковой пластине множества транзисторных структур одновременно (планарная технология). Найдены способы формирования *n*- и *p*-областей в достаточно большом объеме кристалла так, что можно говорить и о трехмерных схемах.

Таким образом, изобретение транзистора можно сравнить с изобретением печатного станка. Ведь как бы мы ни автоматизировали изготовление радиоламп или механических реле, в результате получим отдельные компоненты, содержащие в лучшем случае две-три радиолампы или реле в одном корпусе, которые затем необходимо соединять в одну схему. Вычислительная машина в таком случае — это десятки тысяч радиоламп. При таком количестве отдельных элементов, каждый из которых может выйти из строя, время на ремонт может во много раз превысить время полезной работы. Развитие групповой технологии изготовления транзисторов доказало, что по надежности можно сравнить одну радиолампу с целой пластиной, содержащей десятки тысяч транзисторов. По существу, стало безразлично, сколько транзисторных структур, соединенных в одну схему, изготовлено на пластине. И по надежности, и по стоимости изготовления их можно рассматривать как один элемент. Таким образом, имея небольшое число физических узлов, ЭВМ получила практически неограниченное число логических элементов. Счет пошел на корпуса микросхем, с «печатного станка» стали сходить целые ЭВМ.

Во что превратился первый транзистор? И в уникальный прибор, выполненный на одной пластине диаметром 76 мм, включающий графический видеопроцессор для дисплея и оперативную память в 3,5М байта, и в микросхемы, содержащие до 50 млн. транзисторов, выполняющие сложные математические операции, и в проект трехмерного компьютера.

А начиналось все 40 лет назад...

## Компьютерная технология обучения в средних специальных учебных заведениях

В конце прошлого года введен новый перечень специальностей среднего специального образования, согласованный с Госпланом СССР. Таким образом, перестройка деятельности среднего специального образования начата с главного определяющего участка, так как именно специализация связывает процесс подготовки кадров с заказчиком, в данном случае с конкретным производством.

Новый перечень специальностей призван устранить сложившиеся диспропорции и недостатки в подготовке кадров, предполагает применение новых технологий обучения, переработки программно-методической документации (квалификационных характеристик, учебных планов, программ, учебников).

Среди основных мероприятий перестройки среднего специального образования — всесторонняя компьютеризация учебного процесса и развитие на этой основе новых форм и методов обучения, направленных на повышение качества подготовки специалистов. Группой работников Минвуза СССР, средних специальных учебных заведений и учеными научно-исследовательских институтов разработана концепция компьютерной технологии обучения.

В текущей пятилетке в средних специальных учебных заведениях будет создано примерно 40 тыс. рабочих мест, оборудованных персональными компьютерами. Расширяются работы по организации банков информации, улучшено информационное обслуживание процесса обучения, развернуто формирование технических центров программирования, организован единый фонд алгоритмов и программ средней специальной школы, налажено тиражирование и сопровождение пакетов прикладных программ. Учащимся средних специальных учебных заведений в зависимости

от специальности будет обеспечено 50—100 ч дисплейного времени.

Компьютерная технология обучения будет развиваться на базе единой концепции, объединяющей на системном уровне различные формы применения вычислительной техники в учебном заведении, подготовки специалистов, переподготовки и повышения квалификации преподавательских и руководящих кадров средней специальной школы с учетом развития международного сотрудничества стран — членов СЭВ по приоритетному направлению научно-технического прогресса «Электронизация народного хозяйства».

Компьютерная технология обучения предполагает:

непрерывность применения средств вычислительной техники в течение всего периода обучения;

всесторонний охват учебного процесса;

однотипность и унификацию технического, программного, организационного и учебно-методического обеспечений; тиражирование элементов технологии с целью ее массового распространения в средних специальных учебных заведениях страны;

адаптацию к изменяющимся условиям применения;

высокую дидактическую и экономическую эффективность;

интеграцию образования с производством и наукой на основе новых принципов их взаимодействия и обеспечения опережающей подготовки специалистов.

Будут создаваться методики самостоятельной работы обучаемых с применением ЭВМ, направленные на развитие логического и аналитического мышления, творческих способностей, формирование умений и навыков системного анализа технических, экономических и социальных проблем, находжде-

ние эффективных методов и средств решения задач. Компьютерная технология ориентирована на индивидуализацию обучения в условиях коллективной деятельности в рамках единого учебно-воспитательного процесса. Важно, что компьютерная технология будет совместима с системой непрерывного школьного, профессионально-технического и вузовского образования, обеспечивает преемственность обучения информатике на всех уровнях образования в рамках единой государственной системы подготовки и профессионального роста кадров.

6 Ставится задача полной адаптации компьютерной технологии к индивидуальным особенностям учащихся, обеспечения состоятельности в овладении знаниями, внедрения методов целевой подготовки по индивидуальным учебным планам, обеспечения постоянного обновления и обогащения содержания учебных материалов компьютерной технологии с учетом новейших достижений науки и народнохозяйственной практики. С учетом создания нового поколения учебно-методической документации должны быть созданы и всемерно развиваться гибкие виды и формы учебной деятельности на базе компьютерной технологии обучения в условиях коренной перестройки учебного процесса и структуры подготовки кадров, в том числе и по новым направлениям научно-технического прогресса. При этом будут развиваться новые формы информационного обеспечения учебного процесса, в том числе справочного, библиографического, документального, множительного. Для этого начинается плановая целенаправленная разработка и тиражирование прикладных программ для учебных дисциплин, электронных учебников и программных средств в рамках отраслевых, региональных и центральных банков учебной информации. Разрабатывается комплекс мер, направленных на стимулирование деятельности преподавательского состава по созданию методов и средств компьютерной технологии обучения. Должны быть созданы предпосылки для внедрения методов и средств вычислительной техники и в

тех областях, где использование ЭВМ не является массовым: право, культура, искусство и т. п.

Совместно с общественными организациями развиваются дополнительные виды обучения учащейся молодежи путем организации нетрадиционных его форм: компьютерных клубов и центров, летних лагерей программистов, факультативов, хозрасчетных курсов и др.

Технология компьютерного обучения в средних специальных учебных заведениях будет развиваться в трех направлениях:

использования ЭВМ в качестве средства обучения в составе автоматизированных обучающих систем, предназначенных для организации учебного процесса и управления познавательной деятельностью;

обучения вычислительной технике, принципам ее функционирования, программированию, разработке аппаратного и программного обеспечения;

обучения применению ЭВМ в технических, экономических, общественных, естественных и гуманитарных дисциплинах, в том числе в составе автоматизированных систем различного назначения.

Интеграция этих направлений должна осуществляться в условиях непрерывного применения средств вычислительной техники в течение всего периода обучения и включать органическое соединение обучения с учебно-производственной работой и привития обучаемым навыков самостоятельной учебной деятельности, в том числе навыков постановки проблемы, выбора и ранжирования целей и задач этой деятельности. Для их обеспечения необходимо создание интегрированных банков данных по различным областям знаний с обеспечением свободного доступа к информационной базе, периодическое обновление и модификация содержания обучения с учетом современного уровня и перспектив развития научно-технического прогресса, оптимальное сочетание базовой и специальной подготовки учащихся.

Для осуществления непрерывности подготовки учащихся средних специальных учебных заведений и совершен-



ствования их квалификации вводятся три уровня базовой подготовки (общий, первый и второй) и два уровня специальной подготовки по шести приоритетным направлениям.

Общий уровень базовой подготовки обеспечивается для групп на базе VIII класса, обучающихся по школьной программе «Основы информатики и вычислительной техники», усиленной для отдельных специальностей, а для групп на базе X класса — изучением предмета «Основы практического использования вычислительной техники и работы с персональными компьютерами».

Первый уровень базовой подготовки предусматривает изучение нового (дополнительного) курса «Вычислительная техника и программирование».

Второй уровень наряду с вышеуказанным курсом предполагает введение нового — «Математические методы в технических и экономических расчетах на ЭВМ».

Специальная подготовка предусматривается по следующим разделам:

«Микропроцессорная техника» (вводится предмет «Основы микропроцессорной техники» — первый уровень, «Применение микропроцессорных средств для автоматизации оборудования» — второй);

«Автоматизация проектирования» (вводятся предметы «Основы автоматизации проектирования» — первый уровень, «Системы автоматизированного проектирования» — второй);

«Автоматизация управления и информационные системы» (вводятся предметы «Основы подготовки, передачи и хранения информации» — первый уровень, «Информационные системы в народном хозяйстве» — второй);

«Автоматизированные системы управления производством» (вводятся предметы «Основы автоматизации производства» — первый уровень, «Автоматизированные системы управления производством» — второй);

«Автоматизированные системы управления технологическими процессами» (вводятся предметы «Основы автоматизации управления технологическими процессами» — первый уровень, «Автоматизированные системы управления

технологическими процессами» и «Программирование технологических процессов» — второй);

«Автоматизированные системы научных исследований» (вводятся предметы «Основы автоматизации научных исследований» — первый уровень, «Автоматизированные системы научных исследований» — второй).

Учитывая важность международного сотрудничества стран — членов СЭВ в этой области, достигнута договоренность о координации работ по разработке содержания и методов обучения информатике учащихся, их подготовке в области применения вычислительной техники, созданию, тиражированию и сопровождению учебных программных средств, созданию специализированных учебных кабинетов для обучения учащихся и повышения квалификации преподавателей средних специальных учебных заведений.

В соответствии с этими задачами можно выделить три группы методов применения ЭВМ в обучении.

В первую группу входят методы развития у обучаемых навыков алгоритмизации решения задач и формирования на этой основе логического и системного мышления. Вычислительная техника при этом становится педагогическим средством, которое совершенствует процесс познания изучаемого объекта или явления. Компьютер при этом должен быть рабочим инструментом, не требующим профессионального знания языков программирования и специализированных навыков работы на нем.

Вторая группа методов включает обучение с помощью моделей, адекватно отражающих функционирование реальных объектов и сущность изучаемых явлений. К этой же группе можно отнести игровые методы активного обучения, дающие навыки принятия коллективных решений на основе анализа альтернативных вариантов.

Третья группа методов связана с обучением применению автоматизированных систем различного назначения: АСУ, АСУТП, САПР и др.

Обучение технике программирования предусмотрено во всех группах, поскольку компьютер становится интеллек-

туальным партнером человека. В связи с этим целесообразно при разработке автоматизированных систем и учебно-методических материалов оптимально сочетать интеллектуальные возможности человека и машины.

При внедрении компьютерной технологии обучения выделяются следующие основные тенденции. Увеличивается объем и повышается качество самостоятельной работы обучаемых, которая все больше становится контролируемой и документированной.

В массовом масштабе развивается имитационное, математическое и физическое моделирование как самостоятельный вид учебных занятий, расширяются игровые формы обучения и на их основе — коллективные формы учебной деятельности.

8

В перспективе на основе плановой подготовки учебно-методического обеспечения ЭВМ и создания централизованных баз знаний будут развиваться различные формы обучения, предусматривающие тиражирование и распространение компьютерной технологии обучения. Дополнительные дидактические возможности предоставляет интеграция существующих и перспективных средств вычислительной техники. Компьютерная технология позволяет ориентировать курсовые и дипломные работы на конкретные разработки программных средств, создание фрагментов баз данных и знаний, что обеспечивает развитие информационных систем.

В условиях компьютерной технологии обучения учебный процесс, по крайней мере в ближайшее время, сохранит свои традиционные формы. При этом урок как форма учебной деятельности будет выполнять еще более ответственные функции, главным образом как средство мотивации, обобщения и выработки целостного представления о данной учебной дисциплине, ее связи с другими учебными дисциплинами, ее значении в формировании специалиста. Вместе с тем увеличится объем учебной нагрузки на практических, лабораторных и других видах занятий, относящихся к активным формам учебной деятельности.

Роль преподавателя в условиях

компьютерной технологии обучения будет оставаться ведущей. Изменение содержания его труда обусловлено тем, что управление самостоятельной работой обучаемых становится важным видом преподавательской деятельности, широкое информационное обеспечение учебного процесса на основе актуализации баз данных и знаний требует от него постоянного обновления учебного материала. При широком распространении целевой подготовки учащихся на основе новых учебных планов и программ работа преподавателя приобретает характер наставничества.

В свою очередь, для успешной реализации компьютерной технологии обучения требуется значительное расширение профессиональной подготовки преподавателей. Методологическим принципом повышения квалификации преподавателей в области компьютерной технологии должна быть опережающая подготовка по перспективным направлениям развития народного хозяйства и социальной сферы.

Средства обеспечения компьютерной технологии обучения подразделяются прежде всего на технические средства комплектации кабинетов информатики, вычислительные лаборатории и дисплейные классы, которые обеспечивают проведение практических и лабораторных занятий с использованием автоматизированных обучающих систем в рамках базовой подготовки.

Специализированные вычислительные лаборатории предназначены для проведения практических и лабораторных занятий по предметам специального цикла (микропроцессорной технике, автоматизации производства, проектированию и т. д.).

Средства вычислительной техники могут входить в состав оборудования лабораторий и кабинетов, обеспечивая оперативную обработку экспериментальных данных, решение задач и упражнений, курсовое и дипломное проектирование. Программируемые микрокалькуляторы должны использоваться для индивидуальных массовых расчетов. Ими комплектуются учебные кабинеты, лаборатории.

При формировании вычислительной

базы средних специальных учебных заведений следует обеспечивать совместимость средств вычислительной техники различных уровней, соответствие современному уровню ее развития, ориентацию на младшие модели СМ ЭВМ, ПЭВМ на основе микропроцессоров КП580 и К1810, наиболее доступные для техникумов; необходимо широко использовать выпускаемые КУВТ-86, КУВТ-87, КУВТ «Корвет», а также объединение ПЭВМ в локальные сети.

Программное обеспечение включает штатное, поставляемое промышленностью, и прикладное, разрабатываемое сторонними организациями или учебными заведениями.

В связи с оснащением средних специальных учебных заведений ПЭВМ особое значение приобретает освоение учащимися навыков работы в среде операционной системы. В средних специальных учебных заведениях используются микроДОС, ОС-1800, ОС-1810 (для ЭВМ на основе МП КР, К1810) или ОС ДВК, РАФОС (для ЭВМ на основе МП К1801), системы на основе микро ЭВМ «Электроника-60».

Психолого-педагогическое обеспечение будет также поддерживаться программными средствами текстового характера, рекомендательного и инструктивного вида материалами.

Для практической реализации компьютерной технологии обучения в настоящее время в средней специальной школе введен ряд новых специальностей в области вычислительной техники и автоматизированных систем. Специальности в большей степени приобрели межотраслевой характер, что значительно расширяет сферу деятельности будущих специалистов.

Организованное повышение квалификации преподавателей средних специальных учебных заведений по вычислительной технике проводится с 1 сентября 1985 г. Для этих целей используются 106 факультетов (институтов) повышения квалификации преподавателей Минвуза СССР, минвузов союзных республик и других министерств и ведомств, имеющих в своем подчинении высшие учебные заведения.

Система непрерывной подготовки уча-

щихся средних специальных учебных заведений в области применения вычислительной техники регламентируется комплексом учебно-методических документов и рекомендаций, разработанным НИИ ВШ совместно с техникумами. В них определены требования к базовой и специальной подготовке учащихся различных специальностей, установлены объем и содержание подготовки на всех этапах обучения, даны рекомендации по оборудованию специализированных учебных лабораторий.

Элементы содержания подготовки учащихся техникумов и училищ в области применения вычислительной техники на уровне учебных дисциплин и их разделов представляют собой учебные модули, каждый из которых с течением времени может совершенствоваться или заменяться. Такой динамизм обеспечивает возможность совершенствования в соответствии с изменяющимися условиями и требованиями научно-технического прогресса.

С целью совершенствования подготовки специалистов в области использования вычислительной техники министерство осуществило комплекс мероприятий.

В учебные планы и программы дисциплин внесены дополнения, направленные на усиление подготовки учащихся в области вычислительной техники и различных областях ее применения (автоматизированные системы, гибкие производственные системы, системы автоматизированного проектирования, автоматизации на основе микропроцессорной техники и др.). Существенно расширен объем практических занятий, направленных на привитие прочных навыков владения ЭВМ.

Кроме того, средним специальным учебным заведениям предоставлено право корректировать программы, дисциплины, определяющие специальную подготовку, в том числе в области вычислительной техники, объемом до 15 % времени от общего числа занятий, определенных действующими планами.

Признано целесообразным:

для каждого уровня базовой и специальной подготовки специалистов в области использования вычислительной

техники разработать учебно-методические комплексы, включающие типовые программы учебных дисциплин, соответствующие учебники и учебные пособия, сборники задач и практических занятий, типовые методики проведения лабораторных работ, рекомендаций по оборудованию соответствующих учебных лабораторий и кабинетов;

один раз в пять лет пересматривать и уточнять квалификационные требования к базовой и специальной подготовке специалистов в области использования вычислительной техники с внесением соответствующих изменений в программы, учебную и учебно-методическую литературу;

10

разработать и в последующем раз в пять лет пересматривать квалификационный справочник, в котором установить уровни подготовки специалистов в области применения вычислительной техники для каждой специальности;

разработать и раз в пять лет пересматривать методику, позволяющую определять возможности отдельных отраслей и их учебных заведений в целом по подготовке специалистов различного уровня в области использования вычислительной техники в зависимости от оснащения вычислительной техникой.

Утверждены две программы по обеспечению подготовки, переподготовки и повышению квалификации в двенадцатой пятилетке специалистов с высшим и средним специальным образованием и квалифицированных рабочих в области разработки, создания и использования вычислительной техники, автоматизированных систем, робототехнических комплексов, гибких производственных систем и систем автоматизированного проектирования. Заданиями этих программ в части подготовки кадров по новым направлениям науки и техники намечено в 1986—1990 гг. подготовить 85,4 тыс. человек (в том числе

43,9 тыс. человек — в области вычислительной техники и автоматизированных систем) со средним специальным образованием. Прием в средние специальные учебные заведения на специальности, определенные комплексной программой, в 1990 г. составит 141,2 тыс. человек.

Региональными и отраслевыми учебно-методическими центрами компьютерной технологии обучения являются создаваемые опорные техникумы, которые являются базами повышения квалификации руководящего состава и преподавателей техникумов и училищ региона, организаторами мероприятий по обмену передовым опытом, разработчиками пакетов учебных прикладных программ.

В опорных техникумах создаются методические комиссии преподавателей, которые разрабатывают, обобщают и распространяют методические пособия по проблемам и информации об уже имеющихся разработках. В опорных техникумах в настоящее время создаются учебно-вычислительные центры с соответствующими штатами сотрудников, на которые возлагается эффективная эксплуатация и совершенствование прикладного программного обеспечения и учебно-методических материалов, создание и внедрение АСУ «Техникум», адаптация и внедрение готовых программных продуктов, разработанных сторонними организациями, ведение фондов алгоритмов и программ, их каталогизация.

Нынешний этап перестройки народного образования чрезвычайно важен и интересен. Компьютеризация учебного процесса развивается вширь и вглубь. Только практическими делами, повседневной кропотливой работой всех работников среднего специального образования можно решить поставленные задачи.

А. ГЕЙН, В. ЖИТОМИРСКИЙ, Е. ЛИНЕЦКИЙ, М. САПИР, В. ШОЛОХОВИЧ

## Программно-методический комплекс по курсу школьной информатики

Интересный опыт по организации преподавания машинного варианта курса ОИВТ накоплен в Свердловске, где удалось сконцентрировать усилия всех организаций и учреждений, способных помочь новому делу. О деятельности свердловчан рассказывает эта статья.

Школьный курс «Основы информатики и вычислительной техники» (ОИВТ) вступил в третий год своего существования. Становление любой новой учебной дисциплины сопряжено с немалыми трудностями: необходимо определить содержание предмета, обучить кадры преподавателей, подготовить и выпустить учебную и методическую литературу. В данном случае необходимо также создавать практически полностью отсутствующую в школах техническую базу. Осложняют дело обычные неорганизованность, безответственность, чиновничье равнодушие, когда, по образному выражению А. П. Ершова, «впереди спускаемого на воду еще далеко не достроенного корабля школьной информатики идет огромная, мутная волна дискредитации нового и масштабного дела через халтуру, нерасторопность, невежество, реальную неспособность» [1]. Тем не менее рождение нового предмета состоялось, информатика и компьютеризация все увереннее занимают место в системе современного школьного образования.

Опыт показывает, что более успешно дело идет там, где удается обеспе-

чить согласованные действия органов народного образования, вузов, академических институтов и базовых предприятий при активной поддержке партийных и советских органов. Примером может служить создание в Свердловске сети межшкольных компьютерных центров и постановка преподавания курса ОИВТ по машинному варианту. Областному научно-методическому совету по компьютеризации под руководством академика Н. Н. Красовского на средства свердловских предприятий удалось закупить 275 персональных компьютеров «Роботрон 1715», организовать более 20 межшкольных компьютерных центров. В частности, организован специализированный учебно-производственный комбинат, получивший название городского компьютерного центра (ГКЦ). В четырех компьютерных классах установлены 48 персональных компьютеров, полностью находящихся в распоряжении школьников, изучающих курс информатики или выбравших себе для работы в УПК профиль, связанный с вычислительной техникой. Количество компьютерных классов и их распределение по городской территории обеспечивают реальную возможность работы с ЭВМ каждого ученика IX и X классов в объеме, предусмотренном программой курса ОИВТ.

В период становления нового учебного предмета важно организовать полноценную методическую работу, предо-

ставить каждому учителю возможность получить научную консультацию, обменяться опытом со своими коллегами, обсудить с ними возникающие проблемы. Горно совместно с педагогическим институтом организовало на базе ГКЦ информационно-методический центр, в котором собраны все методические и программные материалы, подготовленные вузами и научными учреждениями города для учителей и школьников, изучающих информатику, еженедельно проводятся консультации для учителей.

12

Преподавание ведется по экспериментальной программе курса информатики (машинный вариант), дополненной порочной разбивкой учебного материала. Программа подготовлена советом по компьютеризации, пединститутом, обложено и разрешена для проведения экспериментальной работы в Свердловской области Минпросом РСФСР. На ее основе к началу 1987/88 учебного года педагогическим институтом подготовлена и выпущена методическая разработка «Основы информатики» [2], содержащая необходимый дидактический материал и описание лабораторных работ. Вместе с методическими материалами во все компьютерные центры переданы и программные средства, обеспечивающие успешное проведение лабораторных работ. Завершается подготовка книги для учителя, содержащей решение многих задач и методические рекомендации по организации занятий.

Таким образом, в Свердловске создан целостный программно-методический комплекс, обеспечивающий учителей и учащихся всем, что необходимо для эффективного освоения курса информатики. Состав и структура комплекса отражают сформированную советом по компьютеризации концепцию реализации компьютерного всеобуча в условиях наличия необходимой технической базы.

Одна из принципиальных установок этой концепции: компьютер для учащегося — рабочий инструмент, используемый в различных сферах деятельности. Школьники должны знать, какие задачи можно решать с помощью ЭВМ, каковы этапы решения этих задач, понимать, на каком этапе при-

меняется ЭВМ. Они должны владеть навыками работы на ЭВМ и уметь анализировать результаты ее применения. Поэтому основной акцент в курсе ОИВТ сделан на освоении учащимися всех этих этапов решения задач на ЭВМ (постановка задачи, построение математической модели, разработка алгоритма решения, составление программы, расчеты на ЭВМ и анализ полученных результатов), самостоятельном решении задач и проведении вычислительных экспериментов. Здесь уместно привести слова из уже цитированной статьи А. П. Ершова: «Я бы хотел всячески защитить идею перехода в информатику от математики, через понятие алгоритма, вырастающего из уже освоенной алгебраической символики, через разработку алгоритма, предшествующего выходу на машину».

Курс ОИВТ делится на две части. В первой рассматриваются основные этапы решения задач на ЭВМ, во второй изучаются основные средства решения задач с помощью ЭВМ (языки программирования, базы данных и другие прикладные программы). Ядром первой части является обучение алгоритмизации. Здесь учащийся должен усвоить следующие основы: понятие об алгоритме как организованной последовательности действий, допустимых для некоторого исполнителя; понятие о способах организации данных и их соответствии способам организации действий. При изучении понятия алгоритма естественно опираться на интуитивные представления учащихся. В то же время школьники в реальной жизни редко встречаются с алгоритмами, предназначенными для автоматических исполнителей. Поэтому формирование понятия «исполнитель алгоритмов» является наиболее сложным. Необходимо постоянно подчеркивать, что приступать к составлению алгоритмов можно только после того, как достаточно четко определены допустимые действия предполагаемого исполнителя. Этим, в частности, объясняется использование нескольких конкретных исполнителей. ЭВМ на этом этапе применяется для имитации работы исполнителей.

На этапе алгоритмизации нами пред-

лагается использовать различные средства описания алгоритмов: словесные и блок-схемы. Словесное описание позволяет более или менее формально изложить алгоритм и делает превращение алгоритма в программу почти автоматическим.

Ядром второй части курса является понятие пакета прикладных программ. Интерпретатор (компилятор) языка программирования также рассматривается как прикладная программа. Изложение теории имеет здесь ознакомительный характер. Изучение материала идет посредством решения конкретных задач.

Цель данной публикации — описать методический компонент программно-методического комплекса по курсу ОИВТ (программному компоненту будет посвящена отдельная статья), методическую разработку [2]. Как уже говорилось, основная цель курса — обучить школьников решению задач с помощью ЭВМ. В соответствии с этим каждое занятие по ОИВТ делится на теоретическую и практическую части. На теоретической части занятия создаются математические модели и алгоритмы для решения задач. В ходе практической работы учащиеся пишут программы и проводят вычислительные эксперименты на ЭВМ.

Методическая разработка содержит теоретический материал и описания лабораторных работ. Изложение теории построено так, что сначала у учащихся формируется общее понятие на основе уже имеющегося жизненного опыта, затем оно формализуется (в первой части — на базе исполнителей), и наконец демонстрируется его применение при решении конкретной задачи. Важно, что эти задачи имеют не только иллюстративную, но и самостоятельную ценность. Мы считаем, что методы, используемые для их решения (в частности, методы вычислительной математики) нужно включать в содержание предмета ОИВТ. Среди них метод Монте-Карло, решение уравнений методом деления пополам, методы оптимального планирования и методы статистических оценок гипотез. Хотя они далеки от стандартного школьного кур-

са, использование ЭВМ позволяет придать им высокий уровень наглядности: с другой стороны, они позволяют продемонстрировать классы задач, наиболее часто решаемых с помощью ЭВМ. Разумеется, полное математическое обоснование этих методов не может быть включено в курс ОИВТ. Для школьников правомерность их использования достаточно убедительно подтверждается вычислительными экспериментами.

Для проверки уровня усвоения материала в каждом параграфе приведены контрольные вопросы. Закрепление теоретического материала достигается, в частности, решением задач (общим числом более 350) четырех основных типов:

1. Каков результат выполнения алгоритма (программы)?
2. Найти ошибку в алгоритме (программе).
3. Составить алгоритм (программу).
4. Построить математическую модель, составить алгоритм, написать программу.

Сразу отметим, что в задачах четвертого типа в равной степени уделяется внимание всем этапам создания математической модели: выделению исходных предположений, на основе которых строится модель, выяснению того, что следует считать исходными данными и результатами, поиску математических соотношений, связывающих исходные данные и результаты. В частности, большинство задач этого типа (в отличие от привычных для школьников текстовых задач традиционных курсов естественного цикла) сознательно недоопределены, чтобы научить школьников выделять исходные данные и результаты. И вообще, задачи, решаемые в курсе ОИВТ, отличаются по характеру от задач из курсов математики, физики, химии и т. п., даже если они похожи по формулировкам. Это обусловлено, во-первых, тем, что результатом выступает алгоритм, а во-вторых, использованием приближенных методов вычислений.

Кроме того, нами проводится принцип подбора задач «с продолжением», т. е. для решения последующей задачи часто требуется знание решения предыдущей. Это учит школьников опи-

раться на уже выполненные этапы решения задачи на ЭВМ, в частности использовать некоторые ранее составленные алгоритмы в качестве вспомогательных (адаптируя их, если нужно, к новой ситуации).

Таким образом, в соответствии с этапами решения задач на ЭВМ задачный материал организован как по вертикали (наличием задач всех указанных типов в каждом параграфе), так и по горизонтали (последовательным расширением постановки некоторых задач от параграфа к параграфу).

14 В первой главе «Знакомство с ЭВМ», носящей вводный характер, на простейших примерах демонстрируются этапы решения задач на ЭВМ, разъясняется понятие математической модели, приводятся первоначальные сведения об ЭВМ. Здесь же дано описание первой лабораторной работы «Первый раз в дисплейном классе», во время которой учащиеся знакомятся с правилами техники безопасности при работе на ЭВМ, получают представление об архитектуре персонального компьютера, осваивают его клавиатуру и простейшие правила редактирования текста на экране, привыкают воспринимать информацию с экрана дисплея. В ходе этой лабораторной работы учащиеся также осваивают режим непосредственных вычислений (фактически используя имитатор исполнителя Вычислитель, о котором будет рассказано позже).

В заключительном параграфе главы I на примере задачи из курса VIII класса подробно разбирается построение математической модели. Объяснение ведется следующим образом.

*Задача.* Тело движется прямолинейно с ускорением  $a$  м/с<sup>2</sup> и начальной скоростью  $v$  м/с. Требуется определить, какой путь пройдет тело за  $T$  секунд.

Из курса физики ученики знают ответ:

$$S = vT + aT^2/2.$$

Для получения этой формулы на уроках физики по сути дела строилась приближенная математическая модель равноускоренного движения (хотя слово «модель», конечно, не употребля-

лось). Нетрудно убедиться в том, что эта модель основана на следующем допущении (см. п. 13 учебника «Физика 8»): если интервал времени разбит на очень большое количество маленьких промежутков, то мы не сильно ошибемся, если будем считать, что скорость тела на каждом из них постоянна, (т. е. движение равномерно) и меняется «мгновенно» в конце каждого промежутка.

Принято считать, что при неограниченном увеличении числа отрезков разбиения мы получим величину перемещения с любой точностью. Фактически это еще одно предположение, которое лежит в основе модели, приводящей к формуле (1). Выполнив на лабораторной работе вычислительный эксперимент, учащиеся с помощью ЭВМ убедятся в том, что чем мельче отрезки разбиения, тем ближе будет результат к значению, полученному по формуле (1).

Далее, руководствуясь схемой построения математической модели, следует выяснить, что считать исходными данными и результатами нашей модели. Ясно, что исходными являются начальная скорость  $v$ , ускорение  $a$ , время движения  $T$ . Результатом, конечно, будет перемещение  $S$ .

Следующая цель — получить математическое соотношение, связывающее исходные данные и результат. Это соотношение будет определяться тем, на сколько частей разбивается интервал времени.

Пусть интервал времени от 0 до  $T$  секунд разбит на  $N$  равных частей. Величина каждой части составляет  $r = T/N$  секунд. По предположению, скорость тела в течение каждого из этих промежутков времени остается постоянной. В течение первых  $r$  секунд тело движется с начальной скоростью  $v_1 = v$  м/с. На следующем отрезке (от  $r$  секунд до  $2r$  секунд) — со скоростью  $v_2 = v_1 + ar$  м/с. В течение третьего промежутка времени скорость будет равна  $v_3 = v_2 + ar$  м/с. Видно, что последовательность  $v_1, v_2, \dots$  является арифметической прогрессией с первым членом  $v$  и разностью  $d = ar$ . Путь, пройденный телом, ищется по формуле



$$S = v_1 r + v_2 r + \dots + v_n r = (v_1 + v_2 + \dots + v_n) r.$$

Воспользовавшись формулой для суммы  $N$  членов арифметической прогрессии, получаем:

$$S = ((2v + d(N-1))N/2)r = ((2v + ar(N-1))N/2)r.$$

Раскроем скобки и подставим  $T/N$  вместо  $r$ :

$$S = \frac{2vNT + aT^2(N-1)}{2N} = vT + aT^2(N-1)/2N = (vT + aT^2/2) - (aT^2/2N).$$

Эта формула и является математическим соотношением, связывающим исходные данные и результат. Построение математической модели закончено. Полученная для  $S$  формула отличается от формулы (1) слагаемым  $-aT^2/2N$ , которое показывает, с какой степенью точности построенная модель описывает равноускоренное движение.

Остальные два этапа решения задачи (создание алгоритма, составление программы и анализ расчетов на ЭВМ) осуществляются во время лабораторной работы.

Для закрепления материала главы 1 учащимся предлагаются следующие задачи:

1. Известна следующая информация о комнате:

в комнате имеются одна дверь и два окна;

пол, потолок, стены, окно и дверь имеют форму прямоугольников с известными сторонами;

под окнами находятся батареи центрального отопления.

Какие из этих сведений существенны для решения каждой из следующих задач:

а) рассчитать количество мела, необходимое для побелки стен и потолка комнаты;

б) рассчитать количество краски, необходимой для покраски пола;

в) рассчитать количество обоев, необходимое для оклейки стен;

г) рассчитать количество древесины,

необходимое для изготовления двери.

2. Для каждой из задач 1а — 1г завершите построение математических моделей.

3. Тело, соединенное пружиной со стеной, оттянуто на  $S$  метров. Какую работу совершила при этом сила натяжения пружины?

При построении для этой задачи математической модели были сделаны следующие предположения: сила натяжения пружины подчинена закону Гука; если все перемещение тела разбить на большое число маленьких промежутков, то можно считать, что сила  $F$  на каждом из них постоянна и меняется «мгновенно» в конце каждого промежутка.

Завершите построение математической модели.

4. Составьте математическую модель для решения следующей задачи.

Плотина прямоугольной формы перегородивает реку. Определить силу давления воды на плотину.

5. Школьник, не знающий теоремы Пифагора, составил следующую математическую модель для определения длины диагонали произвольного прямоугольного стола.

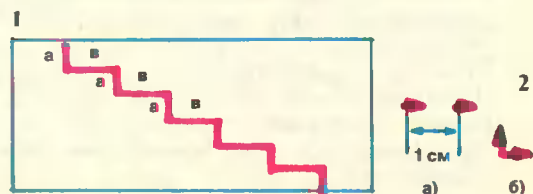
Предположение: поверхность стола считаем прямоугольником; длина диагонали приближенно равна длине ломаной, изображенной на рис. 1, и при неограниченном уменьшении размеров ступенек длина ломаной неограниченно приближается к длине диагонали;

исходные данные: длины сторон стола  $a$  и  $b$ ;

результат: длина диагонали  $d$ ;

связь между исходными данными и результатом:  $d = a + b$  (действительно, длина ломаной в точности равна  $a + b$  независимо от того, сколько в ней ступенек).

Почему получился неверный результат: диагональ прямоугольника равна сумме его смежных сторон?



В главе 2 «Алгоритм и его свойства» даются понятия алгоритма, исполнителя алгоритмов, достижимых целей исполнителя. По мнению авторов, учащиеся должны понимать, что абсолютно строгое определение алгоритма не существует. Понятие алгоритма в информатике является фундаментальным (таким же, какими являются понятия точки, прямой и плоскости в геометрии, пространства и времени в физике, вещества в химии и т. п.). Опираясь на интуитивные представления учащихся, авторы разъясняют понятие алгоритма, как организованной последовательности действий, допустимых для некоторого исполнителя. Под организацией действий в алгоритме понимается то, в каком порядке они выполняются. Основные способы организации действий (ветвления, циклы, вспомогательные алгоритмы) обсуждаются в главах 3—5.

Мы считаем, что на этапе алгоритмизации не следует гнаться за формальностью записи, особенно это касается служебных слов и алгоритмических конструкций. На первых порах лучше записывать служебные слова целиком, в дальнейшем допускаются любые сокращения.

Вот примеры задач, с помощью которых закрепляется понятие алгоритма:

1. Некий злоумышленник выдал следующий алгоритм за алгоритм получения кипятка:

- Налить в чайник воду.
- Открыть кран газовой горелки.
- Поставить чайник на плиту.
- Ждать, пока вода не закипит.
- Поднести спичку к горелке.
- Зажечь спичку.
- Выключить газ.

Исправьте алгоритм, чтобы предотвратить несчастный случай.

2. Определить, для решения какой задачи предназначен этот алгоритм:

- Поставить ножку циркуля в точку А.
- Установить раствор циркуля равным длине отрезка АВ.
- Провести окружность.
- Поставить ножку циркуля в точку В.
- Провести окружность.
- Провести прямую через точки пересечения окружностей.

3. Даны число  $X$  и набор действий: разделить полученное число на 3; умножить  $X$  на 2; сообщить результат; прибавить к полученному числу 4; вычесть из полученного числа 7.

Составьте из этих действий два различных алгоритма. Любой ли алгоритм, составленный из этих действий, можно выполнить? Укажите две различные функции от  $X$ , значения которых вычисляются с помощью алгоритмов, использующих указанные действия.

Первоначально понятие алгоритма разъясняется на примерах алгоритмов, исполнять которые должен человек. Такие примеры позволяют опираться на интуитивные представления учащихся. Однако при составлении программ для ЭВМ те же интуитивные представления становятся помехой, поскольку учащимся с трудом осознается ограниченность набора допустимых действий ЭВМ (и любого конкретного исполнителя). Чтобы преодолеть это препятствие, целесообразно рассмотреть исполнителей с различными наборами допустимых действий (эта идея восходит к Г. А. Звенигородскому). На наш взгляд, учащиеся должны хорошо представлять себе, является ли для данного исполнителя достижимым решение предложенной задачи. Круг допустимых действий должен быть четко очерчен. Обычно в условии задачи либо явно сказано, либо неявно подразумевается, какие действия необходимо исполнить для ее решения. Для школьников такой подход не является непривычным. Например, школьные задачи на построение, как правило, требуется решить с помощью циркуля и линейки. А иногда ставится задача на построение только циркулем или только линейкой. Таким образом учащиеся подводятся к мысли, что различные классы задач требуют различных наборов допустимых действий. Поэтому, чтобы для решения того или иного класса задач можно было применить ЭВМ, необходимо «научить» ЭВМ воспроизводить нужный набор допустимых действий. Мы называем это имитацией исполнителя на ЭВМ. Отметим также, что если в задаче не оговаривается набор допустимых действий, то алгоритм ее решения тривиален.

лен: получить исходные данные; найти решение; сообщить результат.

В нашей методической разработке фигурируют три исполнителя: Чертежник, Вычислитель и Робот-манипулятор. Чертежник рисует линии, передвигаясь по листу бумаги. Его допустимые действия таковы: сделать шаг; повернуть налево; прыгнуть (на один шаг).

Кроме того, Чертежник умеет проверять условие «впереди край». Направление движения Чертежника изображается стрелкой. Пишущий узел находится в начале стрелки. По команде «сделать шаг» он проходит 1 см в направлении стрелки, рисуя линию (рис. 2а); по команде «повернуть налево» он поворачивается на 90 градусов влево вокруг начала стрелки (рис. 2б).

По команде «прыгнуть» Чертежник делает шаг длиной 1 см в направлении стрелки, не рисуя линии. Проверка условия «впереди край» используется при составлении разветвляющихся и циклических алгоритмов.

Один из наиболее широких классов задач, решаемых на ЭВМ,— задачи на вычисления. Для их решения предназначен Вычислитель. Он работает с числами и числовыми переменными. Вот набор допустимых действий этого исполнителя:

1. Запросить исходные данные и обозначить их буквами. Это действие кратко записывается так: «Запросить...», где вместо многоточия ставятся буквы, обозначающие соответствующие данные. Например:

Запросить А

Запросить А, В, С

2. Сообщить результаты вычислений или какой-нибудь текст. Кратко записывается так: «Сообщить...». Здесь вместо многоточия ставятся буквы, обозначающие результаты вычислений, или текст, заключенный в кавычки. Например:

Сообщить Х

Сообщить Х, У

Сообщить «Я работаю Вычислителем!»

3. Обозначить какой-либо буквой значение алгебраического выражения. Краткая запись: «Присвоить... значение...». Вместо второго многоточия пишется выражение, значение которого надо вычислить, а вместо первого многоточия — буква, которой Вычислитель

обозначит это значение. Например:

Присвоить С значение  $A+B$

Присвоить Х значение  $2С$

4. Проверить соотношение между числами или значениями алгебраических выражений ( $<$ ,  $>$ ,  $=$ ,  $\neq$ ,  $\leq$ ,  $\geq$ ).

5. Обозначить буквой случайное число:

Присвоить С значение СЛЗ (а)

По этой команде Вычислитель присвоит переменной С случайное значение из интервала от  $-a$  до  $a$  (распределение равномерное).

6. Запросить таблицу:

Запросить таблицу А (23, 24)

По этой команде Вычислитель запросит последовательно элементы таблицы А, стоящие в первых 23 строках и 34 столбцах.

7. Сообщить таблицу:

Сообщить таблицу А (23, 34)

По этой команде Вычислитель сообщит последовательно элементы таблицы А, стоящие в первых 23 строках и 34 столбцах.

Робот-манипулятор появляется при изучении способов организации данных. Он может перемещаться перед прямоугольным стеллажом, в ячейках которого находятся ящики с однотипными деталями (микросхемами, транзисторами и т. д.). Робот-манипулятор предназначен для поиска нужных деталей. Укажем список допустимых действий Робота-манипулятора.

1. Запросить наименование детали.

2. Переместиться к соседней справа (слева, сверху, снизу) ячейке стеллажа. Сокращенное обозначение: шаг вправо (влево, вверх, вниз).

3. Переместиться к началу стеллажа, т. е. к левой ячейке нижнего ряда.

4. Переместиться к левой ячейке ряда, в котором Робот находится в данный момент.

5. Взять из ячейки деталь и поместить ее в грузовой отсек.

6. Доставить детали к рабочему месту.

7. Проверить, находится ли в ячейке нужная деталь (например, находится ли в ячейке микросхема серии К580).

8. Проверить, является ли ячейка, напротив которой Робот находится, граничной справа (слева, снизу, сверху).

Преподавателю предлагается строить свои объяснения таким образом, чтобы учащиеся воспринимали исполнителя как сочетание рабочего органа и устройства управления, способного воспринимать алгоритмы. При этом можно считать, что управляющее устройство у них одно и то же, что позволяет единообразно записывать алгоритмические конструкции для всех исполнителей. На теоретической части занятия роль устройства управления играет человек (учитель или ученик), при выполнении лабораторных работ эту роль играет ЭВМ.

Разумеется, набор допустимых действий расширяется постепенно, по мере появления соответствующих потребностей. Например, возможность проверки условий появляется в главе 3 («Ветвления»), а действия для работы с таблицами появляются в главе 6 («Организация данных»).

Каждому из исполнителей присущи свои методические функции. Самый простой — Чертежник; он позволяет наглядно иллюстрировать различные способы организации действий. Кроме того, мы используем Чертежника при разъяснении важного понятия целей, достижимых для исполнителя. Учащиеся должны понимать, что алгоритмы, как правило, служат для решения какой-либо задачи, для достижения какой-нибудь цели. Поэтому при выборе исполнителя для решения некоторого класса задач важно знать, какие цели достижимы тем или иным исполнителем, т. е. какие результаты он может получить с помощью присущих ему допустимых действий. Как правило, задача определения всех достижимых целей исполнителя весьма сложна, однако для Чертежника она решается легко (что и было одной из причин выбора этого исполнителя для учебных целей). Первоначально в главе 2 Чертежник может производить только первые два из его допустимых действий. Легко понять, что в этом случае он может рисовать лишь непрерывные ломаные, у которых длины звеньев целочисленны и все углы прямые. Для того чтобы Чертежник мог рисовать и разрывные ломаные, приходится добавить действие «прыг-

нуть». Этим, в частности, иллюстрируется тот факт, что расширение набора допустимых действий влечет расширение класса достижимых целей.

Вычислитель служит для решения «модельных» задач (задач четвертого типа). Кроме того, с его помощью иллюстрируются важные понятия: переменная, присваивание, табличная форма организации данных.

Работа с таблицами становится особенно наглядной благодаря Роботу-манипулятору. Реализация этого исполнителя на ЭВМ позволяет учащемуся постоянно наблюдать за изменениями таблицы во время исполнения алгоритмов. Это способствует лучшему пониманию алгоритмов обработки таблиц и формированию представления о таблице и ее элементах.

Для закрепления понятий исполнителя и достижимых целей нами предлагаются, в частности, следующие задачи:

1. Исполнитель умеет: умножать число на 2; увеличивать число на 1.

А. Составьте для этого исполнителя алгоритм получения числа 100 из единицы.

Б. Сколько действий в самом коротком из таких алгоритмов?

2. Составьте алгоритм нахождения центра тяжести фигур на рис. 3 с помощью карандаша и линейки.

3. Злоумышленник поменял местами действия в алгоритме вычисления среднего арифметического из квадратов трех чисел:

Присвоить  $a$  значение  $(a^2 + b^2 + c^2) / 3$ .

Запросить  $a, b, c$ .

Сообщить «Среднее арифметическое квадратов равно».

Сообщить  $a$ .

Восстановите правильный порядок действий.

4. Какую задачу решит Вычислитель, выполнив следующий алгоритм:

Запросить  $a, d, n$ .

Присвоить  $S$  значение  $(a + d(n - 1))n / 2$ .

Сообщить  $S$ .

5. Какие из этих линий может нарисовать Чертежник (рис. 4)?



a)



б)

}



a)



б)

6. Даны три листа бумаги. Исполнитель берет лист, разрезает его на четыре части и кладет их обратно. Какое количество листов может получиться в результате его работы?

7. Какими допустимыми действиями вы снабдили бы следующих исполнителей:

а) продавца;

б) регулировщика автомобильного движения на перекрестке;

в) клоуна в цирке?

(Эта задача предполагает коллективное обсуждение, в ходе которого будут выявлены наиболее существенные допустимые действия.)

8. Можно ли создать исполнителя, который все может?

9. Даны длины сторон треугольника  $a, b, c$ .

А. Составьте для Вычислителя алгоритм, выполняя который, он сообщит полупериметр треугольника, его площадь и радиус вписанной окружности.

Б. Составьте алгоритм решения той же задачи, в котором используются только четыре переменные.

В. Составьте алгоритм решения той же задачи, в котором используются только три переменные.

В главах 3, 4 и 5 рассматриваются основные способы организации действий: ветвления, циклы и вспомогательные алгоритмы. Мотивацией для их введения служат разнообразные «жизненные» примеры. Запись этих конструкций несколько отличается от принятой в [3]. Как было сказано выше, одно из отличий заключается в том, что мы не «канонизируем» ключевые слова (если, то, иначе, пока, конец цикла и т. п.). Ученики могут писать их полностью, или сокращать, или вообще заменять какими-либо удобными для них знаками. Главное — они должны понимать суть используемой ими конструкции и роль каждого ключевого слова. Разумеется, при переходе к работе с исполнителями, имитируемыми на ЭВМ, появляется необходимость записывать синтаксические конструкции раз и навсегда установленным образом. По существу, в этом и состоит переход от этапа алгоритмизации к этапу работы на ЭВМ. Мы считаем это важной

пропедевтикой введения языков программирования. Кроме того, имитация исполнителей на ЭВМ произведена таким образом, что от учеников не требуется заучивание ключевых слов. Все используемые ключевые слова расположены на функциональных клавишах, значения которых по желанию ученика выводятся на экран в специальной строке.

В методической разработке ветвление записывается одним из двух способов — либо в полной форме:

Если Q, то:

P1

P2

...

T1

Иначе:

T1

T2

...

Конец ветвления

либо в неполной форме:

Если Q, то:

P1

P2

...

Конец ветвления.

Здесь Q — условие, P1, P2, ... — действия, которые выполняются в случае истинности условия Q, а T1, T2, ... — действия, которые выполняются, если это условие ложно. Как только исполнитель выполнил одну из последовательностей действий P1, P2, ... или T1, T2, ..., ветвления заканчивается, т. е. исполнение продолжается с действия, следующего за словами «конец ветвления».

Похожим образом оформляются циклы — если действия P1, P2 ... надо повторять, пока истинно некоторое условие Q, то используется следующая запись:

Пока Q, повторять:

P1

P2

...

Конец цикла.

Эта запись означает, что исполнитель сначала проверяет, истинно ли условие Q. Если да, то совершаются действия P1, P2, ... (последовательность этих действий называют телом цикла), после чего условие Q проверяется снова и т. д. Если же Q ложно, испол-

нитель переходит к действию, записанному после строки «Конец цикла». Видно, что слова «Конец цикла» играют здесь ту же роль, что и слова «Конец ветвления» в записи развилки: они позволяют избежать двусмысленности в записи алгоритмов.

Важно разъяснить учащимся, что не любое условие можно записывать в качестве  $Q$  в ветвлениях и циклах. Исполнитель должен уметь проверять, выполняется ли это условие. Иными словами, с каждым исполнителем связывается определенный набор условий, которые он умеет проверять. Добавляя к набору допустимых действий набор проверяемых условий, мы тем самым расширяем понятие исполнителя.

20

Начиная с третьей главы естественно возникает еще одна форма записи алгоритмов — блок-схемы. Их методические достоинства очевидны. Сначала блок-схемы используются для наглядного представления ветвлений и циклов. Затем (и это становится основной формой их использования) — при построении алгоритмов методом пошаговой детализации.

Приведем примеры задач, предназначенных для закрепления понятия ветвления:

1. Проснувшись утром, школьник почувствовал недомогание. Находившийся рядом злоумышленник тут же составил для него следующий алгоритм:

Измерить температуру.  
Если температура выше  $36,6^\circ$ , то:  
Вызвать врача.  
Пойти в школу.

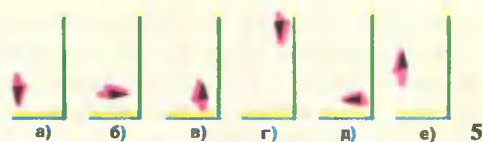
Несмотря на недомогание, школьник исправил этот алгоритм, добавив всего две строки. Какие строки добавил школьник?

2. Для решения какой задачи предназначен следующий алгоритм? Укажите, какое условие в нем проверяется, какие действия совершаются, если условие выполнено, и какие — если условие не выполнено.

Присвоить  $X$  значение суммы углов  $A$  и  $C$  четырехугольника  $ABCD$ .

Присвоить  $Y$  значение суммы углов  $B$  и  $D$  четырехугольника  $ABCD$ .

Если  $X=Y$ , то:



Построить срединный перпендикуляр к отрезку  $AB$ .

Построить срединный перпендикуляр к отрезку  $BC$ .

Найти пересечение построенных перпендикуляров.

Иначе:

Сообщить «Построение невозможно»

Конец ветвления.

3. Запишите в виде алгоритмов правила определения знака:

а) произведения двух действительных чисел;

б) суммы двух действительных чисел.

4. Что нарисует Чертежник, выполнив следующий алгоритм из исходных положений, показанных на рис. 5?

Если впереди край, то:

Повернуть налево.

Конец ветвления.

Если впереди край, то:

Повернуть налево.

Сделать шаг.

Повернуть налево.

Сделать шаг.

Иначе:

Сделать шаг.

Повернуть налево.

Сделать шаг.

Повернуть налево.

Сделать шаг.

Повернуть налево.

Сделать шаг.

Конец ветвления.

5. Составьте алгоритм, с помощью которого Чертежник отойдет от края листа, если первоначально он находится в одном из следующих положений:

а) слева от Чертежника находится край листа и Чертежник не находится в углу листа;

б) Чертежник находится в углу листа и впереди его — край листа (где второй край — справа или слева — неизвестно);

в) Чертежник находится в углу листа и слева от него край листа (где второй край — впереди или сзади — неизвестно);

г) Чертежник находится в углу листа и неизвестно, с какой стороны от него края листа;

д) Чертежник находится у края листа, неизвестно, в углу или нет.

6. В записи алгоритма вычисления значения выражения  $(x^2 - 5x + 5) / (x^6 - 4x^2 + 3)$  злоумышленник одно действие поставил не на свое место. Вот как стал выглядеть алгоритм:

Запросить  $x$ .

Если  $x^6 - 4x^2 + 3 = 0$ , то:

Сообщить «При таком  $x$  значение выражения не определено».

Иначе:

Присвоить  $y$  значение  $(x^2 - 5x + 5) / (x^6 - 4x^2 + 3)$ .

Конец ветвления.

Сообщить «Я нашел значение  $y$ . Вот оно!».

Сообщить  $y$ .

Исправьте алгоритм.

7. Даны координаты двух точек в прямоугольной системе координат. Составьте для Вычислителя алгоритм, с помощью которого он сможет определить, какая из этих двух точек находится дальше:

а) от начала координат;

б) от окружности данного радиуса с центром в начале координат.

8. Составьте математические модели и алгоритмы решения следующих физических задач:

А. Имеется электрическая цепь, составленная из двух сопротивлений, соединенных последовательно или параллельно; определить тип соединения, если известны величины каждого из сопротивлений и сопротивление всей цепи.

Б. Два шара движутся прямолинейно навстречу друг другу; для каждого шара определить, остановится ли он после удара, продолжит движение в том же направлении или начнет двигаться обратно.

9. Составьте математические модели и алгоритмы решения следующих задач по химии:

А. В смеси водорода и кислорода произошла реакция образования воды. Какие вещества и в каком количестве получились в результате реакции?

Б. В колбу с едким натром добавили некоторое количество 15 %-го раствора соляной кислоты. Какие вещества и в каком количестве получились в результате реакции?

В лабораторной работе 3 «Решение



задач с использованием разветвляющихся алгоритмов» разбирается задача о движении тела при наличии трения.

**Задача.** На горизонтальной шероховатой поверхности покоится тело. На него начинает действовать горизонтально направленная сила (рис. 6). Какой путь пройдет тело под действием этой силы за  $T$  секунд?

Математическая модель и алгоритм решения этой задачи составляются учащимися заранее. Вот примерный вид математической модели.

Предположения: поскольку поверхность негладкая, на тело действует сила трения. Считаем, что сила трения покоя равна силе трения скольжения и равна произведению коэффициента трения на ускорение свободного падения (которое считается равным  $9,8 \text{ м/сек}^2$ ) и на массу тела.

Исходные данные: масса тела  $M$ , сила  $F$ , коэффициент трения  $K$  и время  $T$ .

Результат: величина пути  $S$ .

Математические соотношения, связывающие результат с исходными данными:

$$S = \begin{cases} \frac{F - 9,8KM}{2M} T^2, & \text{если } F > 9,8KM \\ 0, & \text{если } F \leq 9,8KM \end{cases}$$

На лабораторной работе учащиеся проводят вычислительный эксперимент с программой для Вычислителя, составленной в соответствии с этой моделью. Цель лабораторной работы — объяснить учащимся, что такое вычислительный эксперимент, каковы его преимущества по сравнению с «натурным» экспериментом.

Учащиеся вводят в ЭВМ программу для Вычислителя и запускают ее несколько раз, увеличивая значение силы  $F$  (начиная с 0). Сравнивая результаты работы ЭВМ при различных значениях  $F$ , они видят, что при малых значениях  $F$ , они видят, что при малых значениях путь равен 0 (тело неподвижно), а затем увеличивается пропорционально  $F$ .

Затем им предлагается провести такие же расчеты, меняя значения массы тела и коэффициента трения.

Делается вывод, что важным преимуществом использования ЭВМ является возможность замены физического эксперимента вычислительным экспериментом. Для физического эксперимента нужно брать разные предметы и разные поверхности (где их столько взять?) и каждый раз с новой силой двигать предмет по поверхности. А на ЭВМ можно легко имитировать эксперимент при любых значениях  $M$ ,  $F$ ,  $K$  и  $T$ .

Важно и другое: вычислительные эксперименты позволяют определить границы применимости построенной математической модели, т. е. выяснить, для каких значений исходных данных результаты согласуются с физической реальностью. Учащимся предлагается подставить вместо  $F$  отрицательное число, достаточно большое по модулю. Получится  $S=0$ . Получается, что если тянуть тело в одну сторону, то оно движется, а если с такой же силой тянуть его в противоположном направлении, то стоит на месте. Учащиеся подвоятся к пониманию того, что построенная модель отражает реальный процесс далеко не при всех значениях аргументов. Так, при ее построении молчаливо предполагалось, что сила  $F$  не может быть отрицательной. Кроме того, использовались законы Ньютона, которые справедливы лишь для небольших скоростей. Значит, и сила  $F$  не может быть слишком большой. Для переменных  $M$ ,  $K$ ,  $T$  также есть естественные границы: их значения должны быть положительными.

Итак, вычислительный эксперимент показал учащимся необходимость уточнения математической модели. В нее надо внести следующие ограничения на исходные данные:  $F \geq 0$ ,  $M > 0$ ,  $K > 0$ ,  $T > 0$ , а также какое-нибудь разумное ограничение на  $F$  сверху, например  $F < 1000 \text{ Н}$ .

Соответствующим образом учащиеся меняют алгоритм решения задачи и проводят расчеты по соответствующей программе.

В главе 4, посвященной циклической форме организации действий, разби-

раются две «модельные» задачи. Первую из них можно назвать «экологической».

**Задача.** Расположенный на берегу реки металлургический завод осуществил сброс в реку вредных веществ, и их концентрация в реке резко увеличилась. С течением времени эта концентрация, естественно, уменьшается. Требуется сообщить, каков будет уровень загрязнения реки через сутки, двое суток и т. д.— до тех пор, пока концентрация не станет меньше предельно допустимой.

Сначала строится математическая модель изучаемого явления. Специалистами-экологами установлена следующая приближенная закономерность: в каждом конкретном случае можно указать такое число  $K > 1$ , что концентрация примесей  $C$  уменьшается в  $K$  раз за сутки. При этом коэффициент  $K$  зависит от района, где протекает река, типа примесей и т. д. Значение  $K$  можно узнать из соответствующего справочника. Эта закономерность принимается в качестве исходного предположения для нашей математической модели.

Исходными данными будут начальная концентрация  $C$  вредных веществ в реке, предельно допустимая концентрация и коэффициент  $K$ . Результат — последовательность значений концентрации вредных веществ через сутки, двое суток и т. д. Связь между исходными данными и результатом дается следующими соотношениями:

$$C_0 = C; C_{n+1} = C_n / K,$$

где  $C_n$  — концентрация вредных веществ через  $n$  суток после сброса.

Исходя из этой математической модели, учащиеся составляют алгоритм решения задачи, а затем, во время лабораторной работы № 4, проводят вычислительный эксперимент.

В этой же главе учащиеся знакомятся с одним из наиболее часто применяемых методов нахождения площадей фигур с помощью ЭВМ — методом Монте-Карло. Очевидно, что реализация метода Монте-Карло требует циклической формы организации действий. Другим важным мотивом выбора этого метода для включения в курс ОИВТ



является возможность продемонстрировать, что математические модели полезны не только для решения задач, далеких от математики, но и для решения сугубо математических задач. Мы объясняем метод Монте-Карло следующим образом.

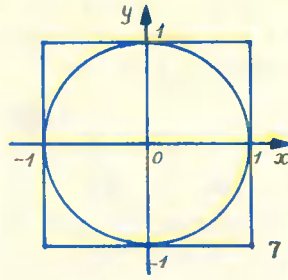
**Задача.** Дана фигура  $F$ . Вычислить ее площадь.

Чтобы построить математическую модель, поместим эту фигуру в квадрат. Будем теперь наугад (как говорят математики, случайным образом) бросать точки в этот квадрат. Естественно предполагать, что чем больше площадь фигуры  $F$ , тем чаще в нее будут попадать точки. Представьте себе квадратный двор и в нем детскую площадку. Каждому ясно, что во время снегопада количество снежинок, попавших на детскую площадку, пропорционально ее площади. Таким образом, можно сделать такое допущение: при большом числе точек, наугад выбранных внутри квадрата, доля точек, содержащихся в  $F$ , приблизительно равна отношению площади  $F$  к площади квадрата.

Исходными данными в модели являются фигура  $F$ , сторона  $a$  квадрата, содержащего фигуру  $F$ , и количество точек  $N$ , которые мы будем случайным образом выбирать внутри квадрата. Результатом является площадь  $S$  фигуры  $F$ . Если через  $M$  обозначить число тех наугад выбранных точек, которые содержатся в фигуре  $F$ , то площадь  $S$  приблизительно равна  $a^2 M/N$ . Разумеется, к связям между исходными данными и результатом следует отнести и математические соотношения, позволяющие определить, попала ли выбранная точка в фигуру  $F$ .

Далее мы предлагаем применить построенную математическую модель для приближенного нахождения площади круга радиуса  $R$ , тем более, что формула площади круга  $S = \pi R^2$  была сообщена учащимся фактически без доказательства.

Пусть для определенности  $R=1$ . На рис. 7 изображен круг радиуса 1, заключенный в квадрат со стороной 2. Таким образом, в качестве фигуры  $F$  выступает круг единичного радиуса,  $a=2$ .



Выбрать точку — это значит задать ее координаты: числа  $x$  и  $y$ . Точка принадлежит квадрату, если  $-1 \leq x \leq 1$  и  $-1 \leq y \leq 1$ . Если  $x^2 + y^2 \leq 1$ , то точка попадает в круг  $F$ , иначе она лежит вне круга. Это и есть математическое соотношение, позволяющее для каждой точки определить, лежит ли она в  $F$ .

На лабораторной работе учащиеся проводят вычислительный эксперимент по нахождению площади круга единичного радиуса. Здесь учащиеся впервые сталкиваются с новой для себя ситуацией, когда при одних и тех же исходных данных могут получаться различные результаты работы программы. Это позволяет наглядно продемонстрировать учащимся понятие вероятностей модели.

Приведем наиболее характерные задачи главы «Циклическая форма организации действий в алгоритмах».

1. «Приключения Тома Сойера» начинаются с того, что тетя Полли зовет Тома:

- Том!
- Нет ответа.
- Том!
- Нет ответа.
- Том!
- Нет ответа...

Составьте алгоритм вызова Тома.

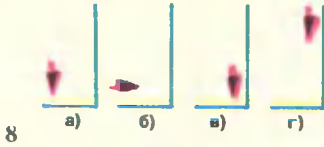
2. Во время большой перемены проголодавшийся школьник зашел в столовую с намерением поесть пирожков. Находившийся рядом злоумышленник тут же посоветовал ему воспользоваться следующим алгоритмом:

Пока не исчезнет чувство голода, повторять:

- Купить пирожок.
- Конец цикла.
- Съесть пирожок.

Сумеет ли школьник поесть пирожков?

3. Что нарисует чертежник, выполнив следующий алгоритм из исходных положений, показанных на рис. 8?



8  
Пока впереди край, повторять:  
Повернуть налево.  
Конец цикла.  
Сделать шаг.

Пока впереди край, повторять:  
Повернуть налево.  
Конец цикла.  
Сделать шаг.

24 4. Чертежник находится у края листа, но не в углу. Составьте алгоритм, выполнив который, он нарисует прямоугольную рамку, отстоящую от края на расстояние одного шага.

5. Составьте для Вычислителя алгоритм, пользуясь которым он запросит последовательность из 10 точек и определит, является ли она:

- арифметической прогрессией;
- геометрической прогрессией.

6. Леспромхоз ведет заготовку деловой древесины. Ее первоначальный объем на территории леспромхоза был равен 123 000 кубометров. Ежегодный естественный прирост составляет 11,5 %. Годовой план заготовки древесины — 2500 кубометров. Каким будет объем деловой древесины на данной территории через год, через два и т. д. — до тех пор, пока он не станет меньше минимально допустимого значения (23 000 кубометров)? Составь-

те математическую модель для этой задачи и алгоритм ее решения.

7 (задача с детективным сюжетом). Каждый день резидент приходит на встречу со своим агентом в случайный момент времени с 11 до 13 часов и ждет его 15 минут. В свою очередь, агент приходит на встречу с резидентом каждый день также в случайный момент времени от 11 до 13 часов и тоже ждет 15 минут. Как часто в течение года встречаются резидент и его агент? Составьте математическую модель и алгоритм решения задачи.

8. Ученик записывает квадратные уравнения вида  $x^2 + px + q = 0$ , выбирая коэффициенты  $p$  и  $q$  случайным образом из отрезка  $[-1; 1]$ . Как часто эти уравнения будут иметь действительные корни? Составьте математическую модель и алгоритм решения этой задачи.

9. Завод, стоящий на берегу реки, сбрасывает в нее ежедневно случайным образом (из-за неисправности очистных сооружений) от 0 до 30 кг вредных веществ. За каждый килограмм сверх 15 завод обязан заплатить штраф 100 руб. Прибыль завода от реализации его продукции — 700 руб. в день. Как часто в течение пятилетки штраф превзойдет прибыль? Рентабелен ли такой завод? Составьте математическую модель и алгоритм решения этой задачи.

Отметим, что задачи 7—9 требуют составления вероятностных моделей.

(Окончание следует.)

## Дидактический материал для текущего контроля знаний

Одним из основных направлений повышения эффективности учебного процесса по курсу ОИВТ является проверка и оценка результатов обучения школьников. При этом ведущая роль принадлежит текущему контролю, который позволяет учителю следить за состоянием знаний школьников на всех этапах изучения учебного материала, оперативно вносить в учебный процесс необходимые коррективы. Кроме того, текущий контроль, охватывая все уровни усвоения учащимися различных тем и разделов курса, наиболее полно осуществляет обучающую функцию.

Предлагаемые дидактические материалы предназначены для проведения учителем текущего контроля по всем разделам курса ОИВТ. Особое внимание в них уделено отработке основных понятий данного учебного предмета, а также навыков применения полученных знаний при решении задач. Включенные в дидактические материалы контрольные задания дифференцированы по уровню сложности, что соответствует разным этапам изучения материала школьниками. Для индивидуального подхода к проверке знаний и умений учащихся, сложность заданий в каждой самостоятельной работе увеличивается от 1 к 6 варианту.

Для избежания разночтения, в материале введены следующие обозначения:

- а) операция умножения \*;
- б) операция деления /;
- в) в записи десятичной дроби используется точка;
- г) условие, записанное вне команды алгоритмического языка, выделяется двумя знаками вопроса ( $A < B$  ?);
- д) не рассматривается тип натуральный.

Время на работу учащихся отводится в пределах 3—10 минут. При ее выполнении школьники должны дать ответы, не переписывая вопросы. По усмотрению учителя, учащимся можно пред-

лагать для выполнения часть вопросов из соответствующего варианта.

### IX КЛАСС

#### Величина

Величина — одно из основных и трудно усваиваемых понятий в информатике.

У учащихся должны быть сформированы и проверены знания по типу величин и их обозначению в алгоритмическом языке.

Необходимо проверить, умеют ли учащиеся: выделить величину из условия задачи, определить тип указанной величины, оценить правильность определения типа величины, выбирать из приведенных значений или самостоятельно указывать значения, допустимые для данной величины, по значению величины устанавливать ее тип, определять для нескольких типов величин соответствующие им значения, определить, является ли величина постоянной или переменной, по величине указать примерную область изменения значений.

#### Самостоятельная работа 1.

##### Вариант 1.1.

1. Укажите тип величины, если ее значение равно:  
а) 25; б) 36.6; в) «нет».
2. Из приведенных значений выберите допустимые значения для величин:  
а) целого типа;  
б) вещественного типа;  
в) литерного типа;  
—5; 3.7; 38; «три»; 20.2; «23»; 14.
3. Определите тип для величин:  
а) вес человека;  
б) количество учащихся в классе;  
в) название дня недели.
4. Для величины **КОЛИЧЕСТВО СТРАНИЦ В КНИГЕ** выберите допустимое значение:

«двести»; 137.5; —18; 243.

5. Приведите несколько допустимых значений для величин:

- напряжение в электросети;
- название времени года;
- площадь фигуры.

В а р и а н т 2.1.

1. Укажите тип величины, если ее значение равно:

- «да»; б) 68; в) 75.4.

2. Из приведенных значений выберите допустимые значения для величин:

- целого типа;
  - вещественного типа;
  - литерного типа;
- «есть решение»; 465; 3.45; 23.79; —47; «986».

3. Определите тип для величин:

- название месяца;
- количество игроков в команде;
- температура человека.

4. Для величины КОЛИЧЕСТВО СТРАНИЦ В КНИГЕ выберите допустимое значение:

—46.3; «девяносто»; 175; 65.1; —187.

5. Приведите несколько допустимых значений для величин:

- название породы дерева;
- высота здания;
- вес человека.

В а р и а н т 3.1.

1. Укажите тип величины, если значение ее равно:

- 48.2; б) «Луна»; в) 29.

2. Из приведенных значений выберите допустимые значения для величин:

- целого типа;
  - вещественного типа;
  - литерного типа.
- 29; 87.05; —946.9; —99; «нет решения»; «68».

3. Определите тип для величин:

- название книги;
- площадь фигуры;
- количество мест в кинотеатре.

4. Для величины КОЛИЧЕСТВО СТРАНИЦ В КНИГЕ выберите допустимое значение:

567; —148; «двадцать пять»; 1481; 314.2; «шесть».

5. Приведите несколько допустимых значений для величин:

- расстояние между городами;

б) название материала;

в) количество учебных предметов.

В а р и а н т 4.1.

1. Укажите тип величины, если значение ее равно:

- «компьютер»; б) 115; в) 0.15.

2. Из приведенных значений выберите допустимые значения для величин:

- целого типа;
  - вещественного типа;
  - литерного типа;
- «55»; 30; —46; —0.126; «Солнце»; 15.003.

3. Определите тип для величин:

- скорость машины;
- название фильма;
- количество учебных предметов.

4. Для величины КОЛИЧЕСТВО СТРАНИЦ В КНИГЕ выберите допустимое значение:

23.9; 293; «сорок»; 16.1; «тридцать пять»; 14.

5. Приведите несколько допустимых значений для величин:

- имя человека;
- скорость ракеты;
- количество приборов в электроцепи.

В а р и а н т 5.1.

1. Укажите тип величины, если значение ее равно:

- 126; б) «слово»; в) 48.37.

2. Из приведенных значений выберите допустимые значения для величин:

- целого типа;
  - вещественного типа;
  - литерного типа;
- 2; «хорошо»; 0.034; —72.4; «42.7»; —81.

3. Определите тип для величин:

- объем жидкости в сосуде;
- номер вагона;
- сорт яблок.

4. Для величины КОЛИЧЕСТВО СТРАНИЦ В КНИГЕ выберите допустимое значение:

«сорок»; —582; «694»; 119.1; 514.

5. Приведите несколько допустимых значений для величин:

- количество дней в месяце;
- название кинотеатра;
- площадь квартиры.

## Вариант 6.1.

1. Укажите тип величины, если значение ее равно:  
а) «147»; б) —27; в) —16.01.
2. Из приведенных значений выберите допустимые значения для величин:  
а) целого типа;  
в) вещественного типа;  
в) литературного типа.  
41; «38.762»; —0.007; —2.4; «27»; 0.
3. Определите тип для величин:  
а) результат прыжка в длину;  
б) год вашего рождения;  
в) марка автомобиля.

4. Для величины **КОЛИЧЕСТВО СТРАНИЦ В КНИГЕ** выберите допустимое значение:

«много»; 1988; —243; 684714950; 99.9; 77.

5. Приведите несколько допустимых значений для величин:  
а) сообщение о результате решения задачи;  
б) показания электросчетчика;  
в) сопротивление проводника.

Письменный ответ ученика может выглядеть следующим образом:

## Вариант 1.1.

1. а) цел; б) вещь; в) лит.
2. а) —5; 28; 14 б) 3.7; 20.2 в) «три»; «23».
3. а) вещь; б) цел; в) лит.
4. 243.
5. а) 127; 220; б) «лето»; «весна»; в) 65.2; 87.34.

## Самостоятельная работа 2.

### Вариант 1.2.

1. Правильно ли определен тип величины?  
а) количество тетрадей тип вещ;  
б) кличка собаки тип лит;  
в) вес контейнера тип цел.
2. Определите, является ли величина постоянной или переменной:  
а) количество дней в неделе;  
б) название естественного спутника Земли;  
в) количество дней в январе месяца;  
г) количество дней в месяце.
3. Напишите заголовок для алгоритма вычисления площади прямоугольного треугольника.

4. Переменные  $x$ ,  $y$ ,  $z$  имеют соответственно тип цел, вещь, лит.

Из приведенных ниже значений укажите запись с допустимыми значениями для  $x$ ,  $y$ ,  $z$ :

- а) 15.3; 6; «пять»;
- б) 38; 26.04; «семь»;
- в) «да»; 18; 10.3.

5. Приведите примерный допустимый интервал значения для величин:

- а) количество дней в месяце;
- б) скорость автомашины.

## Вариант 2.2.

1. Правильно ли определен тип величины?

- а) название хоккейной команды тип лит;
- б) объем шара тип цел;
- в) количество книжных полок тип цел.

2. Определите, является ли величина постоянной или переменной:

- а) стоимость билета в кино;
- б) количество букв в русском алфавите;
- в) температура воздуха;
- г) средняя температура воздуха за прошедшую неделю.

3. Напишите заголовок для алгоритма определения возраста человека.

4. Переменные  $x$ ,  $y$ ,  $z$  имеют соответственно тип цел, вещь, лит.

Из приведенных ниже значений укажите запись с допустимыми значениями для  $x$ ,  $y$ ,  $z$ :

- а) «нет»; 36.8; 27;
- б) 27.6; 13; «один»;
- в) 3; 37.47; «Азия».

5. Приведите примерный допустимый интервал значений для величин:

- а) высота горы;
- б) количество мест в кинотеатре.

## Вариант 3.2.

1. Правильно ли определен тип величины?

- а) название месяца тип цел;
- б) количество рабочих дней в месяце тип цел;
- в) скорость велосипеда тип вещ;

2. Определите, является ли величина постоянной или переменной:

- а) количество минут в часе;
- б) марка автомобиля;

- в) первая космическая скорость;
- г) скорость ракеты.

3. Напишите заголовок для алгоритма определения продолжительности киносеанса.

4. Переменные  $x$ ,  $y$ ,  $z$  имеют соответственно тип цел, вещ, лит. Из приведенных ниже значений укажите запись с допустимыми значениями для  $x$ ,  $y$ ,  $z$ :

- а) 167; «дождь»; 27.81;
- б) 19; —57.3; «ноль»;
- в) «двадцать»; 5; 28.2.

5. Приведите примерный допустимый интервал значений для величин:

- а) влажность воздуха на Земле;
- б) количество этажей в доме.

#### В а р и а н т 4.2.

1. Правильно ли определен тип величины?

- а) количество планет в Солнечной системе тип вещ;
- б) название континента тип лит;
- в) площадь земной поверхности тип цел;

2. Определите, является ли величина постоянной или переменной:

- а) ваше имя;
- б) скорость теплохода;
- в) количество дней в первой декаде месяца;
- г) количество дней в третьей декаде месяца.

3. Напишите заголовок для алгоритма вычисления объема куба.

4. Переменные  $x$ ,  $y$ ,  $z$  имеют соответственно тип цел, вещ, лит. Из приведенных ниже значений укажите запись с допустимыми значениями для  $x$ ,  $y$ ,  $z$ :

- а) «есть»; 35.4; 59;
- б) 0.12; —6; «десять»;
- в) 36; 19.001; «отлично».

5. Приведите примерный допустимый интервал значений для величин:

- а) температура воздуха на Земле;
- б) количество учеников в классе.

#### В а р и а н т 5.2.

1. Правильно ли определен тип величины?

- а) расстояние между городами тип вещ;
- б) марка стали тип цел;
- в) количество букв в слове тип лит.

2. Определите, является ли величина постоянной или переменной:

- а) скорость света;
- б) скорость ветра;
- в) оценка по предмету;
- г) ваша годовая оценка по алгебре.

3. Напишите заголовок для алгоритма вычисления стоимости электроэнергии за прошедший месяц.

4. Переменные  $x$ ,  $y$ ,  $z$  имеют соответственно тип цел, вещ, лит. Из приведенных ниже значений укажите запись с допустимыми значениями для  $x$ ,  $y$ ,  $z$ :

- а) «276»; 37.286; «слово»;
- б) —35; —27.208; «пять»;
- в) —27; 3712.7; «ноль».

5. Приведите примерный допустимый интервал значений для величин:

- а) количество выходных дней в месяце;
- б) количество осадков, выпадающих за месяц.

#### В а р и а н т 6.2.

1. Правильно ли определен тип величины?

- а) масса воды в сосуде тип цел;
- б) количество слов в телеграмме тип цел;
- в) порода собаки тип цел.

2. Определите, является ли величина постоянной или переменной:

- а) частота биения сердца;
- б) количество дней в году;
- в) количество химических элементов в периодической системе Менделеева;
- г) количество химических элементов в периодической системе Менделеева на сегодняшний день.

3. Напишите заголовок для алгоритма вычисления средней скорости тела за три часа.

4. Переменные  $x$ ,  $y$ ,  $z$  имеют соответственно тип цел, вещ, лит. Из приведенных ниже значений укажите запись с допустимыми значениями для  $x$ ,  $y$ ,  $z$ :

- а) 12; «3.3»; «нет»;
- б) —287; —0.56; «287.1»;
- в) «6»; 2.4; «да».

5. Приведите примерный допустимый интервал значений для величин:

- а) скорость движения человека;
- б) количество страниц в книге.

Письменный ответ ученика может выглядеть следующим образом:

## Вариант 1.2.

1. а) нет; б) да; в) нет.
2. а) постоянная; б) постоянная; в) постоянная; г) переменная.
3. алг Площадь (вещ  $a, h, s$ )  
арг  $a, h$   
рез  $s$
4. б) 5. а) [28; 31] б) [0; 1300].

### Команда присваивания

У учащихся должны быть сформированы и проверены знания по синтаксису и семантике данной команды.

Необходимо проверить, умеют ли учащиеся: записывать соотношения между величинами в виде команды присваивания; выделять аргументы, результаты и промежуточные величины; исполнять команду присваивания; определять типы величин в серии команд присваивания; применять команду присваивания при решении задач.

### Самостоятельная работа 3

#### Вариант 1.3.

1. Выпишите номера команд, в записи которых допущены ошибки:  
1)  $x:=a+b$  4)  $a1=7-d$   
2)  $2B:=9$  5)  $2*a:=a+1$   
3)  $y:=\text{«нет»}$  6)  $c:=c+1$
2. Определите тип величины  $D$ , если:  
1)  $D:=A-3.7$  3)  $D:=\text{«да»}$   
2)  $D:=28$  4)  $D:=a/b$
3. Запишите следующее предписание в виде команды присваивания:  
а) увеличить значение  $M$  на единицу;  
б)  $V$  сумма величин  $L$  и  $F$ .
4. Найдите значение переменной  $V$  после исполнения команды присваивания:  
 $V:=(L+3)*2$   
если начальное значение  $L$  равно:  
а) 2; б) 1.5.
5. Найдите значение переменной  $z$  после исполнения серии команд:  
 $a:=8; b:=a*4; z:=b-28$

#### Вариант 2.3

1. Выпишите номера команд, в записи которых допущены ошибки:  
1)  $V:=\text{«ЭВМ»}$  4)  $\text{«ЭВМ»}:=\text{компьютер}$   
2)  $2C:=A+B$  5)  $K=d*a$   
3)  $a-b:=7$  6)  $s:=s-1$

#### 2. Определите тип величины $D$ , если:

- 1)  $D:=K*1.2$  3)  $D:=285$   
2)  $D:=\text{«сок»}$  4)  $D:=k*d$
3. Запишите следующее предписание в виде команды присваивания:  
а) увеличить значение  $s$  в пять раз;  
б)  $z$  сумма (разность) величин  $M$  и  $N$ .
4. Найдите значение переменной  $V$  после исполнения команды присваивания:  
 $V:=(15-L)*4$   
если начальное значение  $L$  равно:  
а) 11; б) 14.5.
5. Найдите значение переменной  $z$  после исполнения серии команд:  
 $a:=5; b:=a*2; z:=a*b$

#### Вариант 3.3.

1. Выпишите номера команд, в записи которых допущены ошибки:  
1)  $V=f+b-5$  4)  $n:=n-2$   
2)  $9d:=139$  5)  $W-D:=L+5$   
3)  $\text{«Ира»}:=\text{имя}$  6)  $z:=a*b-r$
2. Определите тип величины  $D$ , если:  
1)  $D:=\text{«мир»}$  3)  $D:=P-4.28$   
2)  $D:=a-b$  4)  $D:=47$
3. Запишите следующее предписание в виде команды присваивания:  
а) уменьшить значение  $T$  в три раза;  
б)  $R$  произведение величин  $T$  и  $z$ .
4. Найдите значение переменной  $V$  после исполнения команды присваивания:  
 $V:=L*2-7$   
если начальное значение  $L$  равно:  
а) 3; б) 2.5.
5. Найдите значение переменной  $z$  после исполнения серии команд:  
 $c:=10; d:=c/2; z=c-d$

#### Вариант 4.3.

1. Выпишите номера команд, в записи которых допущены ошибки:  
1)  $y:=y/2-y$  4)  $\text{«Мир»}:=\text{кино-театр}$   
2)  $f-1:=f-1$  5)  $z=5*z$   
3)  $\text{теплоход}:=$  6)  $\text{Идом}:=36$   
 $\text{«Пушкин»}$
2. Определите тип величины  $D$ , если:  
1)  $D:=23$  3)  $D:=N*0.32$   
2)  $D:=\text{«есть»}$  4)  $D:=D/3$
3. Запишите следующее предписание в виде команды присваивания:  
а) уменьшить значение  $T$  на шесть;  
б)  $W$  частное величин  $A$  и  $B$ .
4. Найдите значение переменной  $V$  пос-

ле исполнения команды присваивания:

$$V:=L/3+8$$

если начальное значение  $L$  равно

а) 12; б) 1.5.

5. Найдите значение переменной после исполнения серии команд:

$$v:=6; a:=v*v; z:=a-v$$

В а р и а н т 5.3.

1. Выпишите номера команд, в записи которых допущены ошибки:

- 1)  $V:=\langle\text{ГЭС}\rangle$  4)  $s=a*v$   
2)  $d:=8*d-m$  5)  $\langle\text{Москва}\rangle:=\text{город}$   
3)  $2\text{день}:=$  6)  $f+z:=k-w$   
 $\langle\text{вторник}\rangle$

2. Определите тип величины  $D$ , если:

- 1)  $D:=A-0.34$  3)  $D:=\langle\text{белый}\rangle$   
2)  $D:=6234$  4)  $D:=D*D$

3. Запишите следующее предписание в виде команды присваивания:

- а) уменьшить значение  $T$  на  $K$ ;  
б)  $z$  произведение величины  $V$  и 13.

4. Найдите значение переменной  $V$  после исполнения команды присваивания:

$V:=L*2/3$ , если начальное значение  $L$  равно: а) 6; б) 0.75.

5. Найдите значение переменной  $z$  после исполнения серии команд:

$$m:=9; n:=m*2; z:=n/m$$

В а р и а н т 6.3.

1. Выпишите номера команд, в записи которых допущены ошибки:

- 1)  $\text{месяц}:=$  4)  $T+273:=K+M$   
 $\langle\text{май}\rangle$   
2)  $\langle\text{Овод}\rangle:=$  5)  $BC:=DK-M$   
книга  
3)  $\text{дерево}:=$  6)  $S=A*N$   
 $\langle\text{ель}\rangle$

Определите тип величины  $D$ , если:

- 1)  $D:=0$  3)  $D:=2.1+A$   
2)  $D:=\langle\text{шесть}\rangle$  4)  $D:=M$

3. Запишите следующее предписание в виде команды присваивания:

- а) уменьшить значение  $M$  на разность величин  $D$  и  $V$ ;  
б)  $G$  частное величин  $Y$  и  $G$ .

4. Найдите значение переменной  $V$  после исполнения команды присваивания:

$$V:=10*L*L-L$$

если начальное значение  $L$  равно:

а) 5; б) 0.5.

5. Найдите значение переменной  $z$  после исполнения серии команд:

$$x:=18; y:=x/3; z:=y*y+x$$

Письменный ответ ученика может выглядеть следующим образом:

В а р и а н т 1.3.

1. 2; 4; 5.

2. а) вещ; б) цел; в) лит; г) вещ.

3. а)  $M:=M+1$ ; б)  $V:=L+F$ .

4. а) 10; б) 9.

5. 4.

С а м о с т о я т е л ь н а я р а б о т а 4

В а р и а н т 1.4.

1. Исполните серию команд:

$$a:=6; b:=(a+7)*2.$$

2. Запишите следующие предписания в виде серии команд присваивания: увеличить значение  $A$  на 8;

увеличить значение  $A$  в  $V$  раз;

$D$  — сумма значений величин  $A$  и  $V$ .

3. Определите типы величин  $A$ ,  $B$ ,  $C$  в серии команд присваивания:

$$A:=7; B:=A/2; C:=A+B.$$

4. Запишите алгоритм для вычисления площади параллелограмма.

5. После исполнения серии команд

$$c:=d-3; f:=c*5$$

переменная  $f$  получила значение 20. Найдите начальное значение переменной  $d$ .

В а р и а н т 2.4.

1. Исполните серию команд:

$$b:=3; p:=(b+4)*(b/3+2)$$

2. Запишите следующие предписания в виде серии команд присваивания:

уменьшить значение  $K$  на 8;

увеличить значение  $M$  в  $K$  раз;

$A$  — произведение значений величин  $M$  и  $K$ .

3. Определите типы величин  $A$ ,  $B$ ,  $C$  в серии команд присваивания:

$$A:=3; B:=A*7; C:=B/A$$

4. Запишите алгоритмы вычисления площади прямоугольного треугольника.

5. После исполнения серии команд

$$b:=v+3; a:=b-5$$

переменная  $a$  получила значение 11. Найдите начальное значение переменной  $v$ .

В а р и а н т 3.4.

1. Исполните серию команд:

$$d:=8; m:=(d+2)/(d-3)$$

2. Запишите следующие предписания в виде серии команд присваивания:



A — частное значение величин B и C;  
уменьшить значение A на B;  
увеличить значение A на C.

3. Определите типы величин A, B, C в серии команд присваивания:

A:=3.7; B:=A+6.8; C:=A\*B

4. Запишите алгоритм для вычисления объема куба.

5. После исполнения серии команд

P:=B/2; D:=P+4

переменная D получила значение 8. Найдите начальное значение переменной B.

#### Вариант 4.4.

1. Исполните серию команд:

c:=15; t:=(c/3+7)/2

2. Запишите следующие предписания в виде серии команд присваивания:

увеличить значение M на 5;

T — разность значений величин D и M;

увеличить значение T на D.

3. Определите типы величин A, B, C в серии команд присваивания:

A:=5; B:=2/A; C:=3/A+B

4. Запишите алгоритм для вычисления площади равностороннего треугольника.

5. После исполнения серии команд

s:=L-6; Y:=s/3

переменная Y получила значение 24. Найдите начальное значение переменной L.

#### Вариант 5.4.

1. Исполните серию команд:

p:=6; n:=4+p/2-p

2. Запишите следующие предписания в виде серии команд присваивания:

K — произведение значений величин B и Y;

T — разность значений величин K и M;

увеличить значение T на 9.

3. Определите типы величин A, B, C в серии команд присваивания:

A:=4.5; B:=A\*2; C:=B/3

4. Запишите алгоритм для вычисления площади трапеции.

5. После исполнения команд

K:=s\*3; M:=K/4

переменная M получила значение 6. Найдите начальное значение переменной s.

#### Вариант 6.4.

1. Исполните серию команд:

x:=12; y:=(14+x/2)/5+6

2. Запишите следующие предписания в виде серии команд присваивания:

K — частное значение величин D и C;

T — разность значений величин D и K;

M — произведение значений величин T и K.

3. Определите типы величин A, B, C в серии команд присваивания:

A:=4; B:=3/A+2; C:=B\*8-25

4. Запишите алгоритм для вычисления стоимости израсходованной электроэнергии.

5. После исполнения серии команд

A:=8/B; C:=A\*5

переменная C получила значение 40. Найдите начальное значение переменной B.

Письменный ответ ученика может выглядеть следующим образом:

#### Вариант 1.4.

1. 26

2. A:=A+8; A:=A\*B; D:=A+B

3. цел A; вещ B, C

4. алг Упр4 (вещ a, h, s)

арг a, h

рез s.

нач

s:=a\*h

кон

5. 7

#### Команда ветвления

У учащихся должны быть сформированы и проверены знания по синтаксису и семантике команды ветвления (полной и сокращенной формы).

Необходимо проверить, умеют ли учащиеся записывать условие в виде отношений; проверять истинность условия; исполнять команду ветвления; подбирать тестовые значения для проверки команды ветвления; применять команду ветвления (полную и сокращенную форму) при решении задач.

#### Самостоятельная работа 5.

#### Вариант 1.5.

1. Соблюдается ли условие ?  $A > 5$  ? если A равно:

а) —3; б) 8; в) 0; г) 4,9.

2. Соблюдается ли условие ?  $T = \text{«ток»}$  ?  
если  $T$  равно;

а) «кот»; б) «ток»; в) «ТОК»;  
г) «ТОК».

3. Подберите значения величины  $V$ , при которых условия:

а) ?  $V \geq 5.2$  ?; б) ?  $V = \text{«код»}$  ?

соблюдаются; не соблюдаются.

4. Запишите следующее предложение в виде отношения, если необходимо проверить:

а) больше ли значение величины  $A$  числа 5;

б) меньше ли значение величины  $D$  числа 8.35;

в) равно ли значение величины  $M$  «хорошо».

5. Найдите значение величины  $C$  после исполнения команды ветвления:

если  $V \geq 6$

то  $C := 5$

иначе  $C := 12$

все

если значение величины  $V$  равно а) 2;  
б) 16; в) 6.

Вариант 2.5.

1. Соблюдается ли условие ?  $A < B$  ?  
если  $A$  равно:

а) 9; б) —1; в) 2.2; г) 8.

2. Соблюдается ли условие ?  $P = \text{«ЭВМ»}$  ?  
если  $P$  равно:

а) «эвм»; б) «компьютер»; в) «ЭВМ»;  
г) «Э В М».

3. Подберите значения величины  $V$ , при которых условия:

а) ?  $V = 7.5$  ?; б) ?  $V = \text{«ряд»}$  ?

соблюдаются; не соблюдаются.

4. Запишите следующее предложение в виде отношения, если необходимо проверить:

а) меньше ли значение величины  $C$  числа —6;

б) больше ли значение величины  $V$  числа 7.7;

в) равно ли значение величины  $D$  «Саша».

5. Найдите значение величины  $C$  после исполнения команды ветвления:

если  $D \leq -4$

то  $C := 6.5$

иначе  $C := 25$

все

если значение величины  $D$  равно: а) 4;  
б) —4; в) —9?

Вариант 3.5.

1. Соблюдается ли условие ?  $A \geq 8$  ?  
если  $A$  равно:

а) 18; б) 0; в) 8; г) —7.3.

2. Соблюдается ли условие ?  $A = \text{«лето»}$  ?  
если  $A$  равно:

а) «лето»; б) «зима»; в) «лето»;  
г) «ЛЕТО».

3. Подберите значения величины  $V$ , при которых условия:

а) ?  $V \leq -4$  ?; б) ?  $V = \text{«корт»}$  ?

соблюдаются; не соблюдаются.

4. Запишите следующее предложение в виде отношения, если необходимо проверить:

а) больше ли значение величины  $A$  числа 5;

б) меньше ли значение величины  $D$  числа 8.35;

в) равно ли значение величины  $M$  «отлично».

5. Найдите значение величины  $C$  после исполнения команды ветвления:

если  $V \geq 6$

то  $C := 5$

иначе  $C := 12$

все

если значение величины  $V$  равно: а) 2;  
б) 16; в) 6.

Вариант 4.5.

1. Соблюдается ли условие?  $A \leq 6$  ?  
если  $A$  равно:

а) —6.8; б) 0; в) 6.8; г) 7.8.

2. Соблюдается ли условие?  $A = \text{«МАЙ»}$  ?  
если  $A$  равно:

а) «май»; б) «Май»; в) «МАЙ»;  
г) «МИР».

3. Подберите значения величины  $V$ , при которых условия

а) ?  $V \leq -2.5$  ?; б) ?  $V = \text{«лед»}$  ?

соблюдаются; не соблюдаются.

4. Запишите следующее предложение в виде соотношения, если необходимо проверить:

а) равно ли значение величины  $A$  «три»;

б) меньше ли значение величины  $D$  числа —5.5;

в) больше ли значение величины  $M$  значения величины  $K$ .

5. Найдите значение величины  $C$  после

исполнения команды ветвления:

если  $V \geq -2$

то  $C := V + 1$

иначе  $C := 5$

все

если значение величины  $V$  равно: а) 2;  
б)  $-10$ ; в)  $-2$ .

Вариант 5.5.

1. Соблюдается ли условие?  $A \geq -3.2$ ?

если  $A$  равно:

а) 5; б)  $-3$ ; в)  $-3.2$ ; г)  $-7.4$ .

2. Соблюдается ли условие?  $A = \text{«тир»}$ ?  
если  $A$  равно:

а) «ТИР»; б) «т и р»; в) «три»; г) «Тир».

3. Подберите значения величины  $V$ ,  
при которых условия:

а) ?  $V \leq -1.7$ ?; б) ?  $V = \text{«ПАР»}$ ?  
соблюдаются; не соблюдаются.

4. Запишите следующее предложение  
в виде отношения, если необходимо  
проверить:

а) больше ли значение величины  $A$   
значения величины  $C$ ;

б) меньше ли значение величины  $D$   
значения суммы величин  $V$  и  $C$ ;

в) равно ли значение величины  $M$   
«четыре».

5. Найдите значение величины  $C$  после  
исполнения команды ветвления:

если  $V \leq -4.2$

то  $C := V - 1$

иначе  $C := V + 1$

все

если значение величины  $V$  равно: а) 4.2;  
б)  $-4.2$ ; в)  $-8$ .

Вариант 6.5.

1. Соблюдается ли условие?  $A \leq -7.1$ ?  
если  $A$  равно:

а)  $-7$ ; б) 7.1; в)  $-7.1$ ; г)  $-7.2$ .

2. Соблюдается ли условие?  $A = \text{«три»}$ ?  
если  $A$  равно:

а) «3»; б) «т р и»; в) «ТРИ»; г) «три».

3. Подберите значения величины  $V$ ,  
при которых условия:

а) ?  $V \geq -12.8$ ?; б) ?  $V = \text{«Ира»}$ ?  
соблюдаются; не соблюдаются.

4. Запишите следующее предложение  
в виде отношения, если необходимо  
проверить:

а) равно ли значение величины  $A$   
«257»;

б) меньше ли значение величины  $D$   
значения суммы величин  $P$  и  $T$ ;

в) больше ли значение величины  $M$   
значения разности величины  $L$  и числа 5.  
5. Найдите значение величины  $C$  после  
исполнения команды ветвления:

если  $V \leq -2.2$

то  $C := V + 3$

иначе  $C := V/2$

все

если значение величины  $V$  равно: а) 2.2;  
б)  $-2.2$ ; в) 7.

Письменный ответ ученика может  
выглядеть следующим образом:

Вариант 1.5.

1) а) не соблюдается (нет); б) соблю-  
дается (да); в) не соблюдается (нет);  
г) не соблюдается (нет).

2. а) нет; б) нет; в) нет; г) да.

3. да — а) 5.2; б) код; нет — а) 5;  
б) «КОД».

4. а) ?  $A > 5$ ?; б) ?  $D < 8.35$ ?; в) ?  $M =$   
«хорошо».

5. а) 12; б) 5; в) 5.

Самостоятельная работа 6.

Вариант 1.6.

1. Найдите значение величины  $C$  после  
исполнения команды ветвления:

$C := 2$

если  $V > 5$

то  $C := 10$

все

если значение величины  $V$  равно: а) 8;  
б) 3.

2. Запишите следующее предложение  
в виде отношения, если необходимо  
проверить:

а) в месяце 30 дней?

б) точка с координатами  $(X, Y)$  при-  
надлежит оси абсцисс?

(имена величин выбрать самостоя-  
тельно).

3. Запишите исполнение для команды  
ветвления:

если  $D > 3$

то  $C := 8$

иначе  $V := 12$

все

если значение величины  $D$  равно: а) 5;  
б) 0.

33

4. Запишите команду ветвления для следующей системы:

$$Y = \begin{cases} 2, & \text{если } X > 6 \\ 8, & \text{если } X \leq 6 \end{cases}$$

В а р и а н т 2.6.

1. Найдите значение величины С после исполнения команды ветвления:

$$C := 2$$

если  $V > 5$

$$\text{то } C := 10$$

все

если значение величины В равно а) 8; б) 3.

2. Запишите следующее предложение в виде отношения, если необходимо проверить:

а) год високосный?

34

б) точка принадлежит оси ординат? (имена величин выбрать самостоятельно).

3. Запишите исполнение для команды ветвления:

если  $D \geq 3$

$$\text{то } C := 17$$

$$\text{иначе } V := 9$$

все

если значение величины Д равно: а) 3; б) -3.

4. Запишите команду ветвления для следующей системы:

$$Y = \begin{cases} 6.3, & \text{если } X > 0 \\ 5.5, & \text{если } X \leq 0 \end{cases}$$

В а р и а н т 3.6.

1. Найдите значение величины С после исполнения команды ветвления:

$$C := -5$$

если  $V > 8.2$

$$\text{то } C := 27$$

все

если значение величины В равно: а) 8; б) 25.

2. Запишите следующее предложение в виде отношения, если необходимо проверить:

а) температура человека повышена?

б) день недели среда?

(имена величин выбрать самостоятельно).

3. Запишите исполнение для команды ветвления:

если  $D \leq 12$

$$\text{то } C := -8$$

$$\text{иначе } V := 28$$

все

если значение длины Д равно: а) 6; б) 12.

4. Запишите команду ветвления для следующей системы:

$$Y = \begin{cases} -9, & \text{если } X > -4 \\ 1.7, & \text{если } X \leq -4 \end{cases}$$

В а р и а н т 4.6

1. Найдите значение величины С после исполнения команды ветвления:

$$C := 8$$

если  $V > -6.2$

$$\text{то } C := C + 2$$

все

если значение величины В равно: а) 6.2; б) -6.2.

2. Запишите следующее предложение в виде отношения, если необходимо проверить:

а) человек совершеннолетний?

б) угол острый?

(имена величин выбрать самостоятельно).

3. Запишите исполнение для команды ветвления:

если  $D \leq -3$

$$\text{то } C := -13$$

$$\text{иначе } V := 0$$

все

если значение величины Д равно: а) 0; б) -3.

4. Запишите команду ветвления для следующей системы:

$$Y = \begin{cases} 0.7, & \text{если } X > -3.7 \\ 0, & \text{если } X \leq -3.7 \end{cases}$$

В а р и а н т 5.6.

1. Найдите значение величины С после исполнения команды ветвления:

$$C := 2.6$$

если  $V \leq 0.2$

$$\text{то } C := C + V$$

все

если значение величины В равно: а) 0.2; б) 0.

2. Запишите следующее предложение в виде отношения, если необходимо проверить:

а) длины сторон равны?

б) скорость ракеты больше 1 космической скорости?

(имена величин выбрать самостоятельно).

3. Запишите исполнение для команды ветвления:

если  $D \geq -1,5$   
    то  $C := D + 2$   
    иначе  $V := 6.5$

все  
если значение величины  $D$  равно:  
а) — 2.5; б) 0.

4. Запишите команду ветвления для следующей системы:

$$Y = \begin{cases} X + 1, & \text{если } X \geq -7.5 \\ 18, & \text{если } X < -7.5 \end{cases}$$

Вариант 6.6.

1. Найдите значение величины  $C$  после исполнения команды ветвления:

$C := -2$   
если  $V < C + 1$   
    то  $C := V - C$

все  
если значение величины  $V$  равно: а) —1;  
б) —6.

2. Запишите следующее предложение в виде отношения, если необходимо проверить:

а) сосуд пустой?

б) два человека ровесники?

(имена величин выбрать самостоятельно).

3. Запишите исполнение для команды ветвления

если  $D \leq -0.5$   
    то  $C := D + 1.5$   
    иначе  $V := D/2$

все  
если значение величины  $D$  равно: а) 0;  
б) — 0.5.

4. Запишите команду ветвления для следующей системы:

$$Y = \begin{cases} X * X, & \text{если } X > -0.02 \\ X/3, & \text{если } X \leq -0.02 \end{cases}$$

Письменный ответ ученика может выглядеть следующим образом:

Вариант 1.6.

1. а) 10; б) 2.  
2. а) ?  $M = 30$  ?; б) ?  $Y = 0$  ?  
3. а)  $5 > 3$  (да); б)  $0 > 3$  (нет)  
     $C = 8$                        $V = 12$   
4. если  $X > 6$   
    то  $Y := 2$

иначе  $Y := 8$   
все

35

### 3. ШТИЛЬМАН

## Практикум: программирование на основе алгоритмического языка

С февраля 1986 г. в школе № 314 Москвы для учащихся девятых классов Куйбышевского района проходил практикум по информатике и вычислительной технике. Занятия каждого класса проводились в течение одной недели после основных уроков, поэтому специальных домашних заданий не предусматривалось. Класс делился на две подгруппы; с каждой из них проводилось 4 двухчасовых занятия, включавших теоретическую и практическую части. Все объяснения сопровождалось показом через графопроектор примеров из лабораторных работ, заранее подготовленных

учителем, что существенно повышало интенсивность занятий.

Для проведения практикума по программированию на алгоритмическом языке необходимо специальное программное обеспечение. В связи с тем что на данном этапе такие программы для микро-ЭВМ ДВК-1, которыми оборудован наш кабинет, еще не разработаны, пришлось программировать на производственном языке, интерпретатором которого оснащен ДВК-1; в нашем кабинете это интерпретатор Фокала. Для поддержания связи с алгоритмическим языком использовалось программирование

на структурном каркасе алгоритмического языка с построчным кодированием на Фокале.

Идея метода проста. Большинство пространственных версий Фокала допускает запись комментариев в одной строке с выполняемым оператором (правее его) и в отдельной строке, что позволяет ввести в программу и сохранить в качестве комментариев ее текст на алгоритмическом языке. Для этого вводится понятие базового языка, который является расширением алгоритмического языка за счет операторов ввода — вывода Фокала.

### Практическое занятие 1

36

Учащимся даются краткие сведения по технике безопасности и об основных элементах ДБК-1. При помощи плаката их знакомят с клавиатурой. Даются для записи в тетрадь правила включения ДБК-1 и краткие сведения о Фокале.

Алфавит Фокала состоит из букв, цифр и ограничителей. Буквы — латинский алфавит (в комментариях и текстах примечаний при выводе допустимы русские заглавные и строчные буквы). Цифры — только арабские. Ограничители — знаки арифметических операций (возведение в целую положительную степень  $\wedge$ , умножение  $*$ , деление  $/$ , сложение  $+$ , вычитание  $-$ ), разделители (присваивание  $=$ , пробел  $\_$ , возврат каретки **ВК**), знаки препинания  $.$ ,  $:$ ,  $;$ , скобки  $()$ ,  $[\ ]$ , некоторые управляющие символы (кавычки для печати текстовой информации, возврат каретки и перевод строки **!**, вывод на печать значений всех переменных **⊗**).

Операторы Фокала и таблицу встроенных функций учащимся предлагается записать самостоятельно в течение практикума, используя плакаты, которые расположены в кабине вычислительной техники. Для этой цели резервируется две страницы тетради.

### Встроенные функции Фокала

FSIN(X)	$\sin(x)$
FCOS(X)	$\cos(x)$
FTAN(X)	$\operatorname{tg}(x)$
FASIN(X)	$\operatorname{arcsin}(x)$

FACOS(X)	$\operatorname{arccos}(x)$
FATAN(X)	$\operatorname{arctg}(x)$
FLOG(X)	$\ln(x)$
FLOG10(X)	$\lg(x)$
FEXP(X)	$e^x$
FSGN(X)	знаковая часть числа
FITR(X)	целая часть числа
FABS(X)	$ x $
FRAN(X)	генератор случайных чисел
FSQT(X)	$\sqrt{x}$
FCHR(X1, ..., XN)	ввод и вывод символической информации
FCLK( )	функция считывания таймера
FZCLK( )	функция обнуления таймера

### Операторы Фокала

ASK	ввод
TYPE	вывод
SET	присваивание
COMMENT	комментарий
IF	условный переход
GO	безусловный переход
DO	вызов подпрограммы
FOR	присваивание табличных величин
RETURN	возврат из подпрограммы
QUIT	останов программы
MODIFY	редактирование строки
ERASE	стирание текста программы, значений переменных
WRITE	вывод текста программы на экран

Между оператором и следующим символом всегда должен быть пробел.

Кратко рассказывается о работе ДБК-1 в режимах калькулятора и программном. Даются понятия о представлении чисел в ЭВМ. Учащимся знакомят с операторами TYPE, SET, ASK, COMMENT, QUIT, GO, MODIFY, показывают образец оформления линейной программы. Для этого учитель просит повернуть тетрадь на  $90^\circ$  и разделить лист на три части, например так:

№ строки	Кодирование	Алгоритм
----------	-------------	----------

Учитель вместе с учащимися разбирает задачу нахождения произведения двух чисел. При этом он объясняет, что в ба-

зовом языке заголовков основного алгоритма отличается от заголовков вспомогательных: в нем нет списка формальных параметров, отсутствуют строки ARG и REZ. Типы всех переменных описываются в строке НАЧ.

Итак, в правой части листа пишется алгоритм на базовом языке, затем слева номера строк, и только потом в средней части листа выполняется кодирование. Через графопроектор демонстрируется программа (рис. 1).

1.10 С  
1.20 С  
1.30 ;  
1.40 S  
1.50 ;  
1.60 Q

АЛГ ФОРМУЛА 1  
НАЧ ВЕРН А,В,С  
ASK "ВВЕДИТЕ А,В",А,В  
С=А\*В  
TYPE "С=",С  
КОН

Учитель объясняет, что все изображенные на экране компоненты (алгоритм, номера строк и коды) при работе с ЭВМ должны вводиться в нее. Записанные в строках 1.10 и 1.20 заголовки алгоритма и описание типов переменных кодируются в Фокале буквой С (COMMENT), т. е. рассматриваются как комментарии. Команды ввода ASK и вывода TYPE воспринимаются интерпретатором и кодируются «;» для сохранения отступа от левого края строки, присваивание кодируется буквой S (SET), конец программы — буквой Q (QUIT).

Учитель предлагает записать следующие правила программирования на ДБК-1.

Прежде чем начать исполнять программу, нужно проверить правильность ввода текста при помощи команды W→A (WRITE ALL); если есть ошибки, исправить их, используя команду M→⟨номер строки⟩ (оператор MODIFY); для исполнения программы ввести команду GO; перед вводом текста новой программы стереть предыдущую в памяти машины командой E→A (ERASE ALL), чтобы на первых порах не произошло путаницы программ в памяти ЭВМ (эти четыре команды выполняются машиной сразу после их введения). Следует также помнить: после номера строки (перед кодами), а также после кода оператора должны стоять пробелы.

Прослушав объяснения, учащиеся приступают к практической части. Около каждого ДБК-1 лежит текст лабораторной работы (последнее задание является необязательным).

### Лабораторная работа № 1

#### 1. Подготовка ЭВМ к работе.

Включите тумблеры подачи питания на столе.

Включите дисплей нажатием кнопки СЕТЬ.

Клавиши ДУП, ЛИН, РЕД должны быть нажаты, а остальные отжаты вверх.

Включите клавиши в таком порядке:

ПИТ → ПУСК ← → ПР.

2. Работа в режиме калькулятора (режим прямых команд).

Для вывода результата вычислений на экран необходимо использовать оператор TYPE.

Упражнение. Подсчитать сумму чисел 2 и 3.

Решение.

TYPE 2+3 ВК  
5.0000 ВК

или

TYPE «2+3=», 2+3 ВК  
2+3= 5.0000 ВК

Упражнение. Найдите произведение чисел 7 и 3.

Упражнение. Найдите квадрат числа 5.

3. Работа ЭВМ в программном режиме.

Любая строка программы начинается с номера (от 1.01 до 99.99). Для придания ее записи структурности будем использовать символ «;». Запись каждой строки в общем виде будет выглядеть так: ⟨номер строки⟩,⟨код⟩ \_\_\_\_\_  
\_\_\_\_\_⟨оператор⟩ ⟨операнд⟩

Оператор вывода TYPE

*Упражнение.* Вывести на экран число 3. Решение приведено на рис. 2.

*Упражнение.* Подсчитайте сумму чисел 5 и 9. Результат выведите на экран. Решение приведено на рис. 3.

*Упражнение.* Подсчитайте частное двух чисел 48 и 6. Результат выведите на экран.

Оператор присваивания SET (ключевая буква S)

*Упражнение.* Вычислите и выведите на экран сумму значений двух переменных  $A=2$  и  $B=3$ .

Решение приведено на рис. 4 (здесь и далее нажатие клавиши BK не отмечается, но подразумевается).

Оператор ввода ASK

38 *Упражнение.* Введите значение переменной A и выведите его на экран.

Решение приведено на рис. 5.

При исполнении строки с оператором ASK на экране появляется двоеточие — это ЭВМ приглашает вас ввести значение переменной. Если нужно ввести новое значение переменной, то программу надо исполнить заново. Для этого дайте команду GO.

*Упражнение.* Введите программу (рис. 6).

Проверьте правильность ввода с помощью команды W A. Если есть ошибки, исправьте с помощью команды M (номер строки). Исполните программу для произвольных A и B. Покажите учителю.

*Задание 1.* Напишите программу вычисления значения функции  $y=4x^2+3x+9,5$  и исполните ее на ЭВМ.

*Задание 2.* Напишите программу вычисления площади круга.

## Практическое занятие 2

На нем учащиеся знакомятся с разветвляющимися и циклическими программами, которые кодируются с помощью операторов условного и безусловного перехода IF и GO.

В Фокале имеется лишь арифметический оператор условного перехода IF(A)  $n_1, n_2, n_3$ , где A — арифметическое выражение,  $n_1$  — номер строки, куда следует перейти, если  $A < 0$ ,  $n_2$  — номер строки, куда следует перейти, если  $A = 0$ ,  $n_3$  — если  $A > 0$ . С его помощью кодируются элементарные логические выражения. В приведенном в лабораторной работе упражнении 1 закодирована проверка условия  $x < 0$  (строка 1.40).

2	1.10 ;	TYPE 3	<BK>
3	GO		<BK>
4	1.10 ;	TYPE 5+9	<BK>
	GO		<BK>
5	1.10 S	A=2	
	1.20 S	B=3	
	1.30 S	C=A+B	
	1.40 ;	TYPE "C=",C	
	GO		
6	1.10 ;	ASK A	
	1.20 ;	TYPE A	
	GO		
	1.10 C	АЛГ ФОРМУЛА	
	1.20 C	НАЧ ВЕЩ А,В,С	
	1.30 ;	ASK "ВВЕДИТЕ А,В",А,В	
	1.40 S	C=A*B	
	1.50 ;	TYPE "C=",C	
	1.60 Q	КОН	



Для кодирования команды ветвления использован также оператор GO n (безусловный переход к строке с номером n). Он кодирует слово ИНАЧЕ для обхода второй ветви. Стоящие в строках 1.60 и 1.80 операторы присваивания кодируются оператором SET. Служебные слова ЕСЛИ, ТО, ИНАЧЕ, ВСЕ интерпретатором Фокала не воспринимаются и должны быть «закрыты» для него, т. е. предварены буквой С, как комментарии.

Учащимся напоминают общий поряд-

стоятельно. Учитель выступает в роли консультанта.

### Лабораторная работа № 2

*Упражнение.* Введите в ЭВМ программу вычисления функции

$$y = \begin{cases} x & \text{при } x < 0 \\ 2x & \text{при } x \geq 0 \end{cases}$$

Решение приведено на рис. 7.

Проверьте; если нужно, исправьте и исполните для  $x = -2$  и  $x = 5$ .

```
1.10 С
1.20 С
1.30 ;
1.40 IF (X) 1.50,1.80,1.80;С
1.50 С
1.60 S
1.70 GO 1.90;С
1.80 S
1.90 С
2.10 ;
2.20 Q
```

```
АЛГ ФОРМУЛА
НАЧ ВЕЩ X,Y
ASK X
ЕСЛИ X<0
ТО
Y=X
ИНАЧЕ
Y=2*X
ВСЕ
TYPE "Y=",Y
KOH
```

39

7

док написания программы (алгоритм, нумерация строк, кодирование), а также способы кодирования команд, с которыми они познакомились на первом занятии.

В упражнении 2 рассматривается пример циклической программы; команда повторения записана в строках 1.60—2.10. Она кодируется с помощью оператора IF. Если условие повторения выполнено (здесь  $I - N < 0$ ), то переход выполняется на строку, содержащую НЦ (здесь 1.70), иначе — на строку, следующую после КЦ (здесь 2.20). Слово КЦ кодируется оператором GO (безусловным переходом на строку, содержащую слово ПОКА). Ключевые слова ПОКА, КЦ и НЦ «закрываются» для интерпретатора оператором С.

Порядок выполнения лабораторной работы на этом занятии должен быть следующим. Учащиеся для закрепления навыка работы с клавиатурой вводят и исполняют на ЭВМ первое упражнение. Второе упражнение дается как образец и на машине не исполняется. Дальнейшая работа проводится само-

*Упражнение.* Ознакомьтесь с программой вычисления суммы квадратов N первых натуральных чисел (рис. 8).

*Задание.* 1. Напишите программу для вычисления по формуле

$$y = \begin{cases} |x|, & \text{если } x < 0 \\ x + 2, & \text{если } x \geq 0 \end{cases}$$

Исполните на ЭВМ для  $x = 2$  и  $x = -5$ .

*Задание.* 2. Напишите и исполните на ЭВМ программу для вычисления суммы значений первых n четных натуральных чисел.

*Задание.* 3. Напишите и исполните на ЭВМ программу для нахождения наибольшего из двух чисел.

### Практическое занятие 3

На нем учащихся знакомят с кодированием вспомогательного алгоритма, а также с выводом фиксированного изображения на экран дисплея и работой термопечатающего устройства.

Для учителя поясним, что Фокал не поддерживает вызов процедуры с параметрами. Мы моделируем такой вызов, присваивая аргументам вспомога-

```

8  1.10 C
    1.20 C
    1.30 ;
    1.40 S
    1.50 S
    1.60 IF (I-N) 1.70,2.20,2.20;C
    1.70 C
    1.80 S
    1.90 S
    2.10 GO 1.60;C
    2.20 ;
    2.30 Q

```

```

АЛГ СУМКВ
НАЧ НАТ N, ЦЕЛ S, I
АСК "N=", N
I=0
S=0
ПОКА I<N
НЦ
    I=I+1
    S=S+I^2
КЦ
TYPE "S=", S
КОН

```

40

ного алгоритма значения фактических параметров при входе в него и передавая результаты при выходе. Однако аппарат локальных переменных моделировать простыми средствами затруднительно, поэтому следует обратить внимание учащихся на необходимость совпадения имен параметров и промежуточных переменных в основном и вспомогательном алгоритмах.

Кодирование вспомогательного алгоритма рассматривается на примере из упражнения 1. Эта программа демонстрируется через графопроектор. В строках 1.40—1.60 содержится три вызова вспомогательного алгоритма МАХ с различными значениями фактических параметров. При кодировании первого вызова аргументам M и N алгоритма МАХ присваиваются значения фактических параметров X и Y оператором SET. Оператором DO 2 осуществляется переход к исполнению группы строк 2 (строк, у которых в номере перед точкой стоит 2); в них записан вспомогательный алгоритм МАХ. Заголовок вспомогательного алгоритма со списком параметров, а также списки АРГ и РЕЗ кодируются как комментарии. После вызова в строке 1.40 алгоритм МАХ выполняется со значениями M и N, равными значениям X и Y. По окончании вычислений оператором R (RETURN) в строке 2.50 осуществляется переход к оператору, следующему за DO в строке 1.40. Здесь это оператор присваивания S A=K. С его помощью результат вычислений (значение переменной K) присваивается фактическому параметру A. Коман-

да алгоритмического языка, вызывающая вспомогательный алгоритм МАХ (X, Y, A), «закрита» как комментарий. Вызовы в строках 1.50 и 1.60 кодируются аналогично.

Из-за того что вызов вспомогательного алгоритма кодируется оператором DO, каждый из вспомогательных алгоритмов должен занимать одну группу строк (т. е. в номерах его строк перед точками должны стоять одинаковые цифры). В примере это строки 2.01—2.50. Тем самым возникает ограничение на размер вспомогательного алгоритма: не более 90 строк.

Во второй части лабораторной работы учащимся предлагается вывести на экран дисплея изображение. Имеющиеся в нашем кабинете микро-ЭВМ ДВК-1 не оснащены графической приставкой, поэтому изображение предлагается строить с помощью алфавитно-цифрового дисплея, используя оператор TYPE !. Знак «!» является символом управления курсором и вызывает переход на новую строку при каждом выполнении оператора TYPE.

Порядок выполнения лабораторной работы следующий: упражнение 1 служит образцом и не выполняется на ЭВМ. Все остальные задания предназначены для исполнения на ЭВМ.

### Лабораторная работа № 3

*Упражнение.* Ознакомьтесь с программой для вычисления функции

$$u = \frac{\max(x, y) + \max(x+y, xz)}{\max(0,5, x+z)}$$

(рис. 9). В ней используется вспомогательный алгоритм поиска максимального из двух чисел.

**Задание 1.** Написать и исполнить программу для нахождения наибольшего из трех чисел, используя вспомогательный алгоритм поиска максимального из двух чисел.

Вывод изображения на экран.

**Упражнение.** Сформируйте на экране ромб из звездочек.

Решение приведено на рис. 10.

**Задание 2.** Составьте программу для формирования буквы П на дисплее при помощи оператора TYPE !. Высота буквы должна быть пять строк.

**Задание 3.** Напишите программу для вывода на экран любого (по желанию) изображения.

#### Практическое занятие 4

На этом занятии учащиеся знакомятся с кодированием алгоритмов работы

с табличными величинами. В качестве примера используется программа ТАБЛИЦА из упражнения 1.

Дополнительно к уже известным из учебника информатики вводятся специальные команды для работы с табличными величинами: ASKTAB (ввод), TYPETAB (вывод) и SETTAB (присваивание). Они кодируются с помощью оператора FOR I=m<sub>1</sub>, m<sub>2</sub>, m<sub>3</sub>; здесь m<sub>1</sub> — наименьший номер в таблице (в примере для таблицы A[1:N] m<sub>1</sub>=1), m<sub>3</sub> — наибольший номер (m<sub>3</sub>=N), m<sub>2</sub> — шаг перебора элементов таблицы (в строке 1.40 m<sub>2</sub>=1; в строке 1.50 m<sub>2</sub>=2, перебираются нечетные элементы).

Для учителя отметим, что в результате исполнения оператора FOR все операторы Фокала, расположенные в одной с ним строке правее его, повторяются столько раз, сколько этого требуют параметры m<sub>1</sub>, m<sub>2</sub>, m<sub>3</sub> (после каждого исполнения переменная I увеличивается на m<sub>2</sub>). Поскольку интерпретатор Фокала

41

1.10 C	АЛГ ФОРМУЛА
1.20 C	НАЧ ВЕЩ X,Y,Z,U,A,B,C
1.30 ;	ASK X,Y,Z
1.40 S M=X;S N=Y;DO 2;S A=K;C	MAX(X,Y,A)
1.50 S M=X+Y;S N=X*Z;DO 2;S B=K;C	MAX(X+Y,X*Z,B)
1.60 S M=0.5;S N=X+Z;DO 2;S C=K;C	MAX(0.5,X+Z,C)
1.70 S	U=(A+B)/C^2
1.80 ;	TYPE "U=",U
1.90 Q	КОН
1.95 C	-----
2.01 C	АЛГ MAX(ВЕЩ M,N,K)
2.05 C	АРГ M,N
2.10 C	РЕЗ K
2.15 C	НАЧ
2.20 IF (M-N) 2.25,2.40,2.40;C	ЕСЛИ M<N
2.25 C	TO
2.30 S	K=N
2.35 GO 2.45;C	ИНАЧЕ
2.40 S	K=M
2.45 C	ВСЕ
2.50 R	КОН

1.10 TYPE !," \*"  
 1.20 TYPE !," \* \*"  
 1.30 TYPE !," \*"

распознает операторы по первой букве, ASKTAB, TYPETAB и SETTAB будут выполняться как обычные ASK, TYPE и SET, но столько раз, сколько задано в операторе FOR.

#### Лабораторная работа № 4

*Упражнение.* Ознакомьтесь с программой ввода одномерного массива и вывода его элементов с нечетными номерами.

Решение приведено на рис. 11.

11

1.10 C  
1.20 C  
1.30 ;  
1.40 FOR I=1,1,N;  
  
1.50 FOR I=1,2,N;  
42 1.60 Q

Рассмотренный 8-часовой цикл из четырех занятий позволяет путем решения простых задач органично перейти от безмашинного варианта изучения информатики к практическому использованию ранее полученных значений при программировании на микро-ЭВМ. После практикума у учащихся, как правило, повышался интерес к предмету, у многих появилось желание углубить свои знания алгоритмического языка. Учи-

```
АЛГ ТАБЛИЦА
НАЧ ЦЕЛ N,T, ВЕЩ TAB A[1:N]
  АСК N
  ТУРЕТАВ !,"АС",I,"]=";
  АСКТАВ А[1]
  ТУРЕТАВ !,"АС",I,"]=",А[1]
КОН
```

*Задание 1.* Напишите и исполните программу ввода одномерного массива и вывода количества отрицательных элементов.

*Задание 2.* Напишите и исполните программу ввода одномерного массива и вывода его положительных элементов.

По окончании четвертой лабораторной работы учащиеся знакомят с игровыми и обучающими программами, имеющимися в кабинете, что расширяет их представления о возможностях ЭВМ, повышает эмоциональный настрой.

теля, пройдя практикум вместе со своими учащимися, осваивают метод программирования на структурном каркасе алгоритмического языка, осознают его ценность как эффективного средства для выражения конструкций алгоритмического языка операторами производственных языков. В результате и после практикума, продолжая изучение информатики в безмашинном варианте, некоторые из них выполняют кодирование отдельных алгоритмов, как это делалось во время практикума.

## К сведению читателей

Президиум Верховного Совета СССР образовал союзно-республиканский Государственный комитет СССР по народному образованию, упразднив Министерство просвещения СССР, Министерство высшего и среднего специального образования СССР и Государственный комитет СССР по профессионально-техническому образованию.

Эта информация была опубликована в центральной прессе непосредственно перед сдачей рукописи номера в набор, поэтому внести в нее соответствующие коррективы не представлялось возможным.

## Методика преподавания информатики в компьютерной физматшколе МИЭМ

Физико-математическая школа Московского института электронного машиностроения, одна из старейших вечерних физматшкол (в 1986 г. ей исполнилось 20 лет), накопила богатый опыт в преподавании традиционных предметов естественного цикла: элементарной математики, начал математического анализа и вычислительной математики, физики с элементами радиоэлектроники. В 1982 г. в ФМШ началось изучение курса «Основы программирования для ЭВМ», а с 1985 г. на смену ему в учебных планах всех групп школьников с VIII по X класс пришел новый учебный предмет «Основы информатики и вычислительной техники». Перед школой была поставлена качественно новая задача: на основе усиления межпредметных связей, использования ЭВМ в преподавании математики, введения вычислительного эксперимента в физику поднять обучение на принципиально новый уровень, достигнуть которого традиционными средствами было бы невозможно; прийти к овладению учащимися компьютерно-математической культурой. С 1986 г. в названии физматшколы появилось слово «компьютерная», что явилось отражением новых задач, стоящих перед ней, а также, как нам кажется, перед всеми школами страны.

КФМШ МИЭМ работает в сравнительно благоприятных условиях: институт достаточно хорошо оснащен вычислительной техникой, занимается проблемами автоматизации производства и проектирования и имеет один из крупнейших в стране факультет прикладной математики. В процессе подготовки студентов-математиков на факультете накоплен большой научно-методический опыт в обучении информатике и программированию на базе принципов структурного и доказательного программирования, который используется и при работе со школьниками. Для развертывания работ в области школьной информатики, разработки учебных программных средств для средней и высшей школы на факультете было создано студенческое учебно-научное производственное объединение. В него вошли лаборатория информатики кафедры кибернетики, студенческий научно-производственный отряд (СНПО «Информатика») и компьютерная физматшкола, в которой проходят педагогическую проверку программы, разрабатываемые студентами.

Особенностью курса информатики

в МИЭМ и в физматшколе являются лабораторные практикумы на ЭВМ. Особую роль в постановке педагогически успешного практикума играют персональные компьютеры, имеющие развитые средства графического представления информации. Эти машины институт получил от Академии наук СССР для проведения экспериментов в вузе и в школе. Занятия по информатике на персональных ЭВМ проводятся в лаборатории информатики с инструментальным комплектом учебно-вычислительной техники (КУВТ), а также в экспериментальных учебно-производственных мастерских Калининского района Москвы. Здесь МИЭМ установил школьный КУВТ, организовал методический семинар и обеспечивает межшкольный кабинет учебными программными средствами для проведения занятий по основам информатики и вычислительной техники для учащихся IX и X классов четырех школ этого района.

В настоящее время в КФМШ организованы три курса основ информатики: для трехгодичного потока VIII—X классов (225 ч), для двухгодичного потока IX—X классов (150 ч) и для одногодичного потока (75 ч). В основу планирования материала была положена формула 1:1:1 (3 ч в неделю: 1 ч — теория (лекция), 1 ч — практика (решение задач), 1 ч — лабораторный практикум на ЭВМ), что совпадает с соотношением знаний, умений и навыков, необходимых для грамотного использования ЭВМ при решении практических задач. Курсы имеют целью формирование у учащихся: 1) умений грамотной постановки задач, формализованного описания моделей и методов решения задач; 2) умений строить и описывать алгоритмы решения прикладных задач; 3) умений тщательного планирования деятельности и логического обоснования решений; 4) умений работать с ЭВМ и программами обработки данных общего назначения; 5) умений составлять и проводить отладку на ЭВМ несложных программ и знаний возможностей применения и принципов функционирования современных ЭВМ. Главная научно-методическая цель курса — овладение учащимися техникой доказательных рассуждений в анализе алгоритмов и планов решений задач. Для ее достижения в физматшколе изучают элементы математической логики и техники доказательных рассуждений.

Практикум работы на ЭВМ строится на ба-



Занятие в компьютерной физико-математической школе ведет Д. Федюшин

44

зе учебного программного обеспечения, включающего учебные программы: клавиатурный тренажер, модели управления объектами, обучающие игры; деловые программы: графические редакторы, редактор текстов, электронные таблицы, база данных, база знаний. Главной творческой частью практикума является самостоятельное составление алгоритмов и программ для решения задач на ЭВМ, проводимое с применением математических методов, с обязательной предшествующей постановкой решаемой задачи, описанием метода ее решения и сценария диалога с ЭВМ.

Условно курс «Основы информатики» в КФМШ можно разделить на две части: 1) введение в информатику и программирование и 2) прикладную информатику. В первой части учащиеся в основном работают с готовыми программами, знакомятся с методами составления простейших программ, получают первоначальные знания, умения и навыки, необходимые для решения задач с помощью ЭВМ. Во второй части ЭВМ используется как инструмент для самостоятельного решения прикладных задач из физики, математики, информатики, техники, для постановки компьютерных и вычислительных экспериментов. Действующие в КФМШ трех-, двух- и одногодичные курсы информатики в основном отличаются объемом второй части при практически неизменной первой, которая может рассматриваться как машинный вариант курса «Основы информатики и ВТ» в средней общеобразовательной школе.

В основе курса информатики в компьютерной физматшколе МИЭМ лежат систематические методы составления программ и проверки их правильности.

В действующем пробном учебном пособии «Основы информатики и вычислительной техники» (под ред. А. П. Ершова и В. М. Монахова) используется структурный подход к составлению программ, который состоит в тщательном продумывании логики работы программы до ее написания на языке программирования и в составлении так называемых структурированных алгоритмов с записью их на обобщенном алгоритмическом языке, использующем лексику родного языка:

**Задача** → **Алгоритм** → **Программа**

Такой подход применялся в физматшколе МИЭМ с 1982 по 1985 г. Структурированные алгоритмы легко читаются, по ним можно изучать принципы решения задач. Их «перевод» на языки программирования производится по простым правилам и не представляет сложности для школьников.

При разработке сложных алгоритмов и программ используется так называемая пошаговая детализация алгоритмов, при которой первоначальная задача разбивается на обособленные подзадачи; те, в свою очередь, на еще более простые и т. д.

Однако опыт применения принципов структурного программирования в обучении составлению алгоритмов и программ показал, что в созданные таким образом алгоритмы и программы постоянно проникают некоторые трудноуловимые ошибки, связанные с плохим оформлением результатов задач. Работа таких программ иногда требует использования данных, которые не являются исходными, либо появляются непредсказуемые реакции ЭВМ на ошибочные исходные данные. Такие ошибки обнаруживаются, как правило, уже на этапе активного использования программ в вычислительных эксперимен-

тах и при решении задач на ЭВМ. Необходимость поиска и предотвращения таких ошибок потребовала введения новых принципов составления алгоритмов и взаимоувязывания их с понятиями «задача», «способ решения», «постановка задачи», «метод решения», «сценарий диалога», «план решения», «план изложения» и т. п. Результатом методического поиска стали систематические методы построения алгоритмов и решение задач на основе математических постановок, моделей и методов.

Постановка решения задач и вычислительных экспериментов на ЭВМ, так же как постановка спектаклей и кинофильмов, строится на основе сценария, определяющего роли человека и компьютера при их совместной работе. Сценарий регламентирует: как человек управляет работой машины, как и в каком порядке вводятся исходные данные, как ЭВМ представляет полученные результаты и как реагирует на ошибочные данные или некорректные действия пользователя. Алгоритмы, описывающие логику функционирования ЭВМ, должны полностью соответствовать выбранному методу решения и сценарию. Взаимосвязь и взаимозависимость перечисленных фундаментальных понятий можно проиллюстрировать схемой:

Задачи → Способы — содержательный уровень  
 Постановки → Методы — математический уровень  
 Сценарий → Алгоритмы — алгоритмический уровень

ЭВМ — Программы — компьютерный уровень  
 На основе подобной схемы школьники хорошо усваивают этапы решения задач на ЭВМ; легко дать и классификацию ошибок.

Во время перерыва идет оживленное обсуждение составленных ребятами алгоритмов



Первый тип ошибок — синтаксические — это нарушение правил грамматики используемого языка программирования. Второй — нарушение правил алгоритмов на язык программирования. Третий — несоответствие алгоритма сценарию работы с ЭВМ. Четвертый — ошибки реализации в алгоритмах выбранных методов решений. Пятый — дефекты сценария: непонятные сообщения или результаты, сложность правил ввода исходных данных, отказы или недружественные сообщения при нарушении человеком правил работы. Шестой тип ошибок — принципиальные ошибки в выбранных методах решения. Седьмой — некорректная постановка задачи, в которой в число требуемых данных включено не то, что в действительности требуется, и т. п.

Таким образом, принятый в физматшколе МИЭМ подход к составлению алгоритмов и программ использует принципы систематического конструирования и состоит из следующих этапов:

1. Математическая постановка задачи.
2. Определение метода ее решения.
3. Составление сценария работы с ЭВМ.
4. Построение структурированных алгоритмов.
5. Кодирование программ по описаниям алгоритмов.

6. Ввод и испытания программ.  
 В математической постановке задачи обязательно отражаются:

- 1) все данные (т. е. параметры, значения которых должны быть известны для решения конкретной задачи);
- 2) требуемые результаты (т. е. множество изменяемых величин, значения которых являются результатами решения задачи);
- 3) ограничения на данные, вытекающие,

например, из физического смысла задачи. В этом пункте дается ответ на весьма существенный для дальнейшего решения вопрос: какие значения исходных данных являются, а какие не являются допустимыми;

4) связь между результатами и исходными данными. Эта связь есть не что иное, как математическая модель явления, отражаемого в данной задаче.

Уяснив, таким образом, суть задачи, зафиксировав на бумаге математическую постановку, учащийся выбирает математический метод ее решения. Под методом мы понимаем обобщенный способ решения задач данного класса для всех допустимых исходных данных. Для вычислительных задач метод обычно записывается в виде систем уравнений и неравенств с комментариями и с использованием вспомогательных функций.

Сценарии программ для диалоговых компьютеров — это совокупность всех картинок и текстов, предъявляемых человеку, и совокупность команд, с помощью которых человек управляет работой ЭВМ, а также запросы для ввода исходных данных.

При наличии сценария и метода решения написание алгоритмов у школьников уже не вызывает особых трудностей. Для записи алгоритмов используется алгоритмический язык с добавлением операторов диалогового ввода и вывода (типа «запрос»). Запись алгоритма проводится в две колонки: собственно алгоритм и память. Написанный алгоритм легко кодируется на выбранном языке программирования по правилам соответствия конструкций алгоритмического языка конструкциям языка программирования.

К начальным фундаментальным знаниям в области информатики относится «представление о правильности программы и возможности строгого доказательства ее правильности» (Велихов Е. П. Новая информационная технология в школе. Информатика и образование. 1986. № 1. С. 19). Проверка правильности составленных алгоритмов и программ при таком подходе может проводиться задолго до выхода на ЭВМ. Программы сверяются на их однозначное соответствие описаниям алгоритмов. Алгоритмы проверяются на соответствие сценариям и описаниям выбранных методов решения. Правильность методов сверяется с описанием постановок задач. Практически все ошибки можно выявить за столом или у доски. Описание алгоритмов на алгоритмическом языке и описание сценариев в форме картинок, фрагментов текста — предмет интересных активных дискуссий и даже «мозговых атак» в коллективах учащихся.

Методически занятия по составлению алгоритмов, освоению учащимися алгоритмиче-

ского языка, техники постановок задач и описания сценария диалога строятся в процессе коллективных обсуждений. Задача учителя состоит в выборе исполнителей на ту или иную роль: постановщиков задач, сценаристов, алгоритмистов, программистов — и в руководстве их работой. Первым начинает работу постановщик задач (как правило, это один из лучших математиков в классе или группе). Ему же доверяется определение метода и способа решения, а в случае затруднений эту роль берет на себя учитель. Сценаристами обычно становятся ученики, обладающие развитым воображением. В роли алгоритмистов, как правило, выступают наиболее активные ребята — лидеры класса. Программистами могут быть любые ученики, которые практически без ошибок быстро записывают операторы программ.

Запись постановки задачи, фрагменты сценария, описание метода сохраняются на доске до тех пор, пока они нужны для составления алгоритмов и программ. Ограниченные размеры доски вынуждают экономно формулировать все элементы постановок, задач сценариев и алгоритмов. При этом алгоритмы приходится строить пофрагментно на основе тщательно продуманного плана, который поддается или излагается учителем.

Ошибки в таких коллективно составленных алгоритмах и программах почти отсутствуют. Роль контролера выполняет коллектив учащихся, который нацеливает на поиск ошибок и их исправление. Правильность постановок задач контролируется учителем, правильность методов решения обсуждается учителем с учениками, правильность сценариев — это предмет коллективных обсуждений. Правильность алгоритмов контролируется сценаристами и постановщиками задач.

Выявление ошибок при таком подходе — с тщательной проработкой сценария, строгой постановкой решаемых задач и описанием структурированных алгоритмов — проводится путем логических рассуждений. Цель этих рассуждений — выяснить, даст программа требуемые результаты и сообщения или нет. Всякое отклонение от сценария безоговорочно признается ошибкой. Вносить изменения в сценарий после его утверждения не разрешается. Анализ правильности конечных результатов строится на основе анализа результатов промежуточных вычислений и разборе отдельных случаев. Техника анализа правильности постепенно усложняется и от конкретных примеров смещается к разбору частных случаев постановок задач, далее — к выделению подзадач и анализу правильности их решения, а затем — к анализу простейших циклов и введению индуктивных утверждений. Вершиной в анализе является обсуж-



дение условий завершенности циклов, которые продолжаются до выполнения некоторых условий.

Такая подготовка учащихся позволяет поставить перед ними проблему исчерпывающего анализа правильности алгоритмов и планов решения. Для проверки правильности алгоритмов не обязательно записывать их в форме программ и надеяться на то, что ошибки будут найдены на ЭВМ. Испытания программ на ЭВМ позволяют проверить их только на нескольких частных примерах и никаких гарантий в отсутствии ошибок не дают. Гарантии отсутствия ошибок в программах может дать только исчерпывающий логический анализ правильности алгоритмов и методов решения, положенных в их основу.

Кульминационным пунктом в обучении методам составления алгоритмов и программ является введение строгого и категорического понятия правильности:

1. Алгоритм решения задачи правильный, если он дает требуемые результаты при любых допустимых исходных данных.

2. Программа правильная, если она при любых допустимых исходных данных формирует результаты, строго соответствующие требованиям постановок задач, и ее работа полностью соответствует сценарию при любых действиях пользователя.

Проверка программ на ЭВМ заключается в тщательно спланированных испытаниях на соответствие работы машины сценарию программы. При отсутствии сценария непонятно, как можно осуществить более или менее полную проверку. При наличии сценария проверка состоит в апробации всех основных частных случаев постановок задач, проверке граничных условий, проверке реакций программы на недопустимые данные и недопустимые действия со стороны человека.

Испытания работы программ на ЭВМ целесообразно проводить по принципу «ввел — попробовал». Особенно эффективна такая методика при использовании диалоговых языков типа Бейсик. При применении этой методики испытания программы начинаются сразу же после ввода первых ее частей. Для этого по сценарию и описаниям алгоритмов выделяются те части программ, которые позволяют как можно скорее получать результаты решения подзадач с выдачей этих результатов испытателю. Первыми обычно вводятся операторы вывода сообщений, затем — операторы ввода данных, далее — операторы, реализующие основную часть вычислений или обработки данных. Как правило, при тщательном подборе и планировании ввода и испытаний программ более или менее полные испы-

тания программы в целом могут быть завершены при вводе последнего фрагмента программы.

Однако для сложных задач, допускающих широкую вариацию исходных данных, полную проверку программ для всех данных или действий человека провести невозможно. Испытания программ на ЭВМ вселяют в нас уверенность в правильности их работы, но не могут служить абсолютной гарантией. Полную гарантию правильности программ может дать только строгое доказательство того, что при любых допустимых исходных данных программа формирует результаты, полностью соответствующие требованиям постановок задач, а в более широком смысле — что при любых исходных данных и действиях человека работа ЭВМ будет соответствовать сценарию, который должен быть написан с математической строгостью и однозначностью.

Использование математических методов составления алгоритмов и программ дисциплинирует мышление школьника, помогает ему при решении задач из курсов физики и математики и, самое главное, позволяет осознанно и целенаправленно применять знания, полученные в курсе «Основы информатики и вычислительной техники», для решения практических прикладных задач, оценивать возможность их алгоритмического решения.

Опыт преподавания систематических методов составления алгоритмов и программ студентами факультета прикладной математики учащимся компьютерной физико-математической школы МИЭМ показал, что по сравнению с традиционными эти методы изучаются и усваиваются намного проще и эффективнее. Их изучают на курсах компьютерного всеобуча преподаватели института, учителя средних школ Москвы, преподаватели ПТУ, техникумов, педагогических и технических вузов, проходящие в МИЭМ подготовку по новому предмету «Основы информатики и вычислительной техники».

Результаты педагогического эксперимента по этой методике были использованы авторами при написании конкурсного учебника, занявшего второе место на конкурсе Минпроса в 1987 г.

Желающим подробнее ознакомиться с данной методикой и конкретными примерами ее применения мы рекомендуем статьи в журнале «Квант» (1986, № 10, 11; 1987, № 6); в сборнике «Проблемы компьютерного обучения» (Вып. 2. М.: Знание, 1985); учебное пособие: *Каймин В. А., Питеркин В. М.* Основы информатики и вычислительной техники. М.: МИЭМ, 1985.

А. АРХАНГЕЛЬСКИЙ

## Мир ЭВМ

### Часть вторая

Мы снова возвращаемся к операционной системе RT-11 и жизни пользователя в мире ЭВМ. Оставив читателя наедине с программой DIR, мы надеялись, что он проявит то человеческое любопытство, которое двигает прогресс, и мы рады, если наши ожидания оправдались. Однако для серьезного разговора, который мы начинаем, необходимо кое-что напомнить\*.

### Как мы играли с программой DIR

Играть мы должны были не просто так, а со смыслом. Задача была простая — попробовав все ключи в различных комбинациях, посмотреть на каталог системного тома со всех сторон, ведь пока мы занимаемся глазами системы. Чтобы получить самую полную картину, выведем на бумагу (и сохраним для дальнейшей работы) каталог — полный, с указанием номеров начальных блоков, которые даны в восьмеричной системе счисления, с именами файлов, расположенными в одну колонку, в начале которой указан идентификатор тома и имя владельца:

```
DIRECTORY/PRINT/FULL/BLOCKS/OCTAL/  
COLUMNS:1/VOLUMEID DX0:
```

Именно таким будет результат выполнения этой командной строки, ключи в которой выбраны в соответствии с нашим заказом.

Распечатку каталога удаленных файлов мы уже делали, нам остается только найти ее

\* А для тех, кто не читал «Информатику и образование» в прошлом году, сообщить: аналогами различных версий RT-11 являются операционные системы ОС ДВК, РАФОС, ФОДОС, FOBOS.



в старых бумагах; впрочем, можно и повторить снова:

```
DIRECTORY/COLUMNS:1/OCTAL/PRINT/DE-  
LETED DX0:
```

Для читателя, работающего с КУВТ-86, напомним, что вместо устройства DX0: он должен указать MX0:. Число блоков на томе будет несколько меньше, но это не влияет на ход наших рассуждений.

*Делу время, потехе час*

Который час? Но прежде — какое число? Спросим у системы:  
.DATE

Она ответила:  
?KMON—W—NO DATE

Вот так! Нет даты, и все! Придется сказать ей, какое сегодня число:  
.DATE 29—JUL—87



Система «скушала» нашу команду, поставила точку и ничего не сказала. Но если мы теперь ее спросим:

.DATE  
то получим ответ по существу:  
29—JUL—87

Установим и время.

.TIME 9:10:00

.TIME

9:10:03

Все оказалось очень просто. Однако при повторной загрузке системы дату и время необходимо устанавливать снова и забывать об этом не следует.

#### 4. Против лома нет приема, или Программа DUMP

Настало время познакомиться со вторым «глазом» системы — программой DUMP. Как всегда, мы начнем с истории и узнаем, что обозначает это слово.



Англо-русский словарь дает перевод: «свалка», «куча мусора». Вроде бы это далеко от действительного смысла в нашем случае, но не будем спешить. В первых ЭВМ, не имевших средств отладки программ, программист при сбое имел единственную возможность — распечатать на бумаге содержимое оперативной памяти в двоичных или восьмеричных кодах. Ужасающее количество чисел в этих распечатках использовалось лишь для того, чтобы найти среди них одно неверное, которое «убило» программу. А потом вся бумага шла в корзину и на свалку. Постепенно слову DUMP стали придавать смысл «посмертный снимок памяти», т. е. распечатка содержимого памяти, когда программа «умерла», не работает. Сейчас в таких системах, как RT-11, RSX-11 и др., дампом называют не столько распечатку оперативной памяти, сколько распечатку содержимого запоминающего устройства или файла. Для распечатки содержимого оперативной памяти используется возможность сохранить заданный участок памяти в виде файла на внешнем запоминающем устройстве.

Посмотрим, каким может быть дамп в RT-11. Но сначала зададимся вопросом — что мы должны увидеть: примерный вид выводимой информации, примерное количество бумаги, которое нам может потребоваться, или сколько времени информация будет выводиться на экран дисплея?

Как это можно прикинуть? Например, следующим образом: каждое 16-разрядное слово в восьмеричной системе числения отображается 6 символами, в качестве разделителя используется пробел. Блок содержит 256.\*\* слов, значит, на каждый блок, содержимое которого мы хотим посмотреть, программа DUMP должна выдать не менее 1800 символов, что составляет одну машинописную страницу.

А остальное давайте посмотрим на практике.

Предположим, что имя программы и имя команды совпадают (а в большинстве случаев это так), и введем с клавиатуры команду .DUMP

Как видно, система не все поняла, потому что задала вопрос:

DEVICE OR FILES?

Она спрашивает, что нам нужно показать — содержимое устройства или файла. Это и есть режим подсказки. Конечно, содержимое файла всегда меньше, чем содержимое устройства, на котором он находится, поэтому для пробы выберем файл поменьше и назовем его: DEVICE OR FILES? DX0:DX0FIL.DIR

\*\* Напоминаем, что десятичные числа кончаются точкой, а числа без точки — восьмеричные.



Система «проглотила» команду, и из ТПУ поползла голубая бумага с темно-синими числами.

Для экономии бумаги мы, как правило, не будем приводить в тексте примеров полученных распечаток. Читателю предлагается самому получить нужные.

Разберемся в этих числах.

Во-первых, отметим, что в начале распечатки указана спецификация файла. Это полезно для идентификации распечатки в дальнейшем.

Во-вторых, в начале каждой таблицы указан номер блока; есть и какие-то числа в начале каждой строки. Так как мы распечатывали дампы файла, а номера начинаются с нуля, то это номер блока относительно начала файла. Разгадка чисел в началах строк не составит труда, если сосчитать количество чисел в таблице. Очевидно, это номера байтов. Тогда номера (или адреса) слов вдоль строки должны быть 0, 2, 4, 6, 10, 12, 14, 16 (они не распечатываются). В подтверждение этого в начале каждой строки номер увеличивается на 20.

Но что за красная полоса идет сверху вниз на распечатке?

Приглядевшись, мы поймем, что программа здесь ни при чем, а открыв крышку ТПУ, увидим, что бумага кончается и красная полоса предупреждает об этом.

Взяв новый рулон бумаги и заправив его в ТПУ, попробуем все-таки выводить информацию на терминал.

Станем действовать уже известным способом, учтя, что то, что в программе DIR выполнялось по умолчанию, здесь нужно сказать в явной форме:

```
.DUMP/TERMINAL DX0:DX0FIL.DIR
```

Действительно, на экране точно такая же картинка, какая была на бумаге.

#### 4.1. Где начало, где конец, или Как управлять дампом

Теперь, когда мы научились экономить бумагу, можно немного поиграть и научиться управлять распечаткой.

Во-первых, нам необходимо явно указывать конец выводимого текста. Мы знаем, что минимальный адресуемый элемент данных в системе — блок; уже полученные распечатки подтверждают, что программа DUMP работает с блоками устройства или файла. Конец по-английски — END, номер блока скорее всего будет параметром, в качестве устройства для вывода укажем терминал:

```
.DUMP/TERMINAL/END:6 DX0:RT11SJ.SYS
```

В одном из блоков целые фразы по-английски! Выведем этот текст на бумагу и сохраним для дальнейшей работы.

Итак, заканчивать посередине мы научились. А как начать с нужного блока?

Наверное, «начать» — используем слово BEGIN в качестве ключа:

```
.DUMP/TERMINAL/BEGIN:2 DX0:DX.SYS
? KMON—F—INVALID OPTION FOR PROGRAMM
```

Система ругается. Начать? Примерно такой же смысл имеет слово START — попробуем в качестве ключа его:

```
.DUMP/TERMINAL/START:2 DX0:DX.SYS
```

Вот теперь другое дело — удовлетворенно замечаем мы, увидев на экране желанный текст.

Ну а если нам понадобится распечатать информацию только об одном блоке? Что делать тогда?

А это совсем просто. Мы уже знаем слово «только» и смело можем его использовать:

```
.DUMP/TERMINAL/ONLY:6 DX0:
```

Все, что мы сейчас делали, было управлением процессом вывода распечатки дампа.



BLOCK NUMBER 00005

000/	020104	051117	046440	051511	044523	043516	052040	027124	*D OR MISSING TT.*
	EFT	MF1	LMX	MMA	K/K	KPN	MRP	GPT	
020/	054523	000123	053523	050101	051456	051531	047040	052117	*SYS.SWAP.SYS NOT*
	NK*	BC	M85	L3Y	MLN	MMQ	LT	MSW	
040/	043040	052517	042116	044000	047101	046104	051105	043040	* FOUND.HANDLER F*
	KH2	MY9	J7F	KT2	LT3	LHD	MFU	KH2	
060/	046111	020105	047516	020124	047506	047125	000104	053523	*ILE NOT FOUND.SW*
	LHI	EFU	L*v	EF6	L*N	LUM	A.	M85	
100/	050101	043040	046111	020105	047524	020117	046523	046101	*AP FILE TOO SMAL*
	L3Y	KH2	LHI	EFU	L*.	EF1	LNS	LHA	
120/	000114	041077	047517	026524	026527	051105	047522	020122	*L.?BOOT-W-EERROR *
	A6	JW9	L*w	GJD	GJG	MFU	L*z	EF4	
140/	042522	042101	047111	020107	040510	042116	042514	000122	*READING HANDLER.*
	KCZ	J63	LUA	EFW	JQ2	J7F	KCT	BB	
160/	047515	044516	047524	020122	044506	042514	047040	052117	*MONITOR FILE NOT*
	L*u	K/F	L*.	EF4	K.B	KCT	LT	MSW	
200/	043040	052517	042116	041400	047117	046106	041511	044524	* FOUND.CONFLICTI*
	KH2	MY9	J7F	J.2	LUG	LHF	J0Y	K/L	
220/	043516	051440	051531	042507	020116	050117	044524	047117	*NG SYSGEN OPTION*
	KPN	ML	MMQ	KCO	EF0	L39	K/L	LUG	
240/	000123	000000	000000	000000	000000	000000	000000	000000	*S.....*
	BC								
260/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
300/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
320/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
340/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
360/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
400/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
420/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
440/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
460/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
500/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
520/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
540/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
560/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
600/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
620/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
640/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
660/	000000	000000	000000	000000	000000	000000	000000	000000	*.....*
700/	000000	000000	000000	000000	000000	000000	000000	016300	*.....G.*
								DX	
720/	000000	007354	071677	142302	000224	000000	000000	040142	*..J.?CRD.....E*
		BOT	RT1	1SJ	C.			JLR	
740/	005015	052122	030455	051461	020112	053040	032460	030056	*..RT-11SJ V05.0*
	AXM	MSZ	G4/	MLQ	EFZ	M1H	HTP	G.N	
760/	020060	013440	000000	000000	000000	000000	000000	000000	*0 .....
	EF	C.							



Теперь можно заняться исследованием того, в какой форме информация выводится. Нас будет интересовать, не что обозначает каждое конкретное число, а его форма — длинное оно (т. е. 16-разрядное слово) или короткое (8-разрядный байт), или, например, как относятся друг к другу числа и символы в полученных нами распечатках. Мы уже видели в распечатке два поля — числа в восьмеричной системе счисления и какие-то символы, которые иногда складываются в разумные слова и фразы. Как управлять этой информацией?

О числах догадаться не очень сложно — мы распечатывали информацию по словам, а можно, наверное, и по байтам, задав ключ BYTE. Тогда мы увидим, что символы, которые печатались в правом поле распечатки, — попытка отобразить байты в кодах ASCII. Будем считать это подкаской, как отказать от этого поля распечатки. Попробуем «отрицать» коды ASCII.

.DUMP/TERMINAL/NOASCII DX0:DXFILE.DIR  
A теперь попробуем заставить программу DUMP перевести те же слова в код RADIX-50 и вывести на бумагу (распечатка нам еще пригодится) (см. илл.):

.DUMP/RAD DX0:/START:5/END:7

#### 4.2. Как быть с выходным файлом?

Программа DUMP по умолчанию выводит информацию на устройство печати, но может, как мы убедились, вывести ту же информацию на терминал. Было бы очень полезно, как и в случае с программой DIR, выводить информацию на любое устройство.

Чтобы лучше понять ее возможности, повторим все наши действия, но другим способом. Например, вывод на терминал:

.DUMP/OUTPUT:TT:DX0:DXFILE.DIR

На экране появилась знакомая картинка. Так же можно вывести информацию и на печать, отменив всякие умолчания:

.DUMP/OUTPUT:LP: DX0:DXFILE.DIR

Все повторилось на принтере. Точно таким же образом мы можем вывести информацию на внешнее запоминающее устройство, надо только полностью набрать спецификацию файла:

.DUMP/OUTPUT:DX0:DX0FIL DX0:DX0FIL.DIR

Накопитель пошуршал, и на экране появилась точка. Единственный доступный нам способ узнать, произошло ли что-нибудь в системе — посмотреть справочник DX0:

.DIR DX0:

Знаковая картинка, но с новым файлом DX0FIL.DMP.

TT	.SYS	2P	10—MAR—84
DX	.SYS	3P	10—MAR—84
SL	.SYS	13P	10—MAR—84
LP	.SYS	2P	10—MAR—84
MX	.SYS	10P	10—MAR—84
NL	.SYS	2P	10—MAR—84
RK	.SYS	3P	10—MAR—84
MT	.SYS	9P	10—MAR—84
DP	.SYS	3P	10—MAR—84
LD	.SYS	8P	10—MAR—84
PIP	.SAV	29	06—FEB—84
DUP	.SAV	45	06—FEB—84
DIR	.SAV	19	06—FEB—84
DUMP	.SAV	9	06—FEB—84
DXOFIL	.DIR	2	29—JUL—86
DXFILE.	.DIR	2	12—AUG—86
DXOFIL.	.DMP	11	

19 FILES, 271 BLOCKS

215 FREE BLOCKS

Этот файл текстовый, но посмотреть мы его пока не умеем. Отметим только, что тип файла .DMP программа DUMP задает по умолчанию.

Кроме указанных, есть и другие возможности, которые мы пока показать не можем, но назвать должны.

Как и в случае с получением справочника устройства в виде файла по ключу OUTPUT, мы можем зарезервировать место на носителе для выходного файла, используя совместно с ним ключ ALLOCATE и параметр, указывающий размер будущего файла в блоках, например:

.DUMP/OUTPUT:DX0:DMP1.DMP/ALLOCATE:30. DX0:DX0FIL.DIR

Ключ FOREIGN позволяет считывать содержимое магнитной ленты, записанной в произвольном формате, т. е. не так, как это принято в RT-11. Вот пример его использования:

.DUMP/TERMINAL/FOREIGN MT0:

Так как магнитофон у нас подключен только к «сотке», мы не сможем использовать этот ключ при работе на ДВК. Однако есть варианты систем, в которых магнитофон подключен и к ДВК.

Последний, но самый важный ключ программы DUMP — IGNORE.

Как видно из его названия, он позволяет игнорировать ошибки аппаратуры или носителя информации. К счастью, у нас пока все работает хорошо, но если понадобится, можно ввести командную строку

.DUMP/TERMINAL/IGNORE DX0:

и посмотреть содержимое всего устройства, несмотря на возможные плохие блоки и отдельные неполадки аппаратуры.

Все это интересно. Но читатель может спросить — зачем все это нужно?

Конечно, не зная системы, трудно воспользоваться такой подробной информацией,

12—AUG—86			
SWAP	.SYS	26P	13—JAN—84
RT11SJ	.SYS	73P	10—MAR—84

да еще в таком большом объеме. Однако эта же информация поможет нам при исследовании системы, так что назовем основные формы использования программы DUMP: распечатывая подробную информацию о файле, мы можем использовать полученный текст (в коде ASCII или RADIX-50) или некоторые стандартные формы файлов при определении назначения файла или способа его использования;

можно использовать текстовой файл, полученный с помощью программы DUMP, для поиска определенных кодов в исходном файле с помощью программ редактирования текстов;

если по каким-то причинам испортился каталог тома, то, определив физическое положение файлов на томе, мы сможем восстановить доступ к информации. (При подготовке этой книги из-за неисправности аппаратуры был испорчен каталог у двух дисков, по 20М байт каждый. В результате 80 тыс. блоков информации — это примерно 20 тыс. страниц машинописного текста — стали недоступны, и только с помощью программы DUMP удалось спасти информацию.)

54



На этом можно закончить рассказ о программе DUMP. Остались только некоторые подробности.

### 4.3. Некоторые подробности, или Ключи и сокращения для интерпретатора командной строки

Для вызова программы DUMP нужно набрать на клавиатуре командную строку

```
.RUN DX0:DUMP
```

В ответ увидим звездочку.

Мы уже знаем, что язык CSI значительно беднее KMON, и если мы уже вызвали программу DUMP, то нужно набрать спецификацию выходного файла. Все как для программы DIR. Только если мы не укажем спецификацию выходного файла, то информация выведется на устройство печати, а не на терминал.

Раз мы уже знаем все, то просто приведем таблицу ключей для программы DUMP. А читателю предлагается попробовать проделать все, что мы делали в KMON, но используя ключи CSI.

Ключ в KMON	Сокращение для KMON	Значение для CSI
/ASCII	AS	—
/WORDS	W	W
/BYTES	BY	B
/RAD50	R	X
/ONLY:BLOCK	ON	O:VAL
/START:BLOCK	ST	S:VAL
/END:BLOCK	EN	E:VAL
/IGNORE	IG	G
/FOREIGN	FO	T
/PRINTER	PRI	—
/TERMINAL	TE	—
/OUTPUT:SPF	OU	—
/ALLOCATE:SIZE	ALL	—
/NOASCII	NOAS	N

Для упражнения, а также потому, что они будут нужны нам в дальнейшем, предлагаем читателю вывести на бумагу две распечатки: дамп в 16-разрядных словах и перевод их в код RADIX-50, но без отображения в ASCII;

дамп в байтах с переводом их в код ASCII.

*Продолжение следует*



## «Прозаические редакторы» на ЭВМ

Самонадеянно скажет иной:  
«Сколочу-ка телегу!»  
Но ведь в телеге-то сотня частей!  
Иль не знает он, дурень?  
Гесиод. «Труды и дни».

### Проблема

Хороша Аннушка, да хвалит только бабушка. Работа над текстом является существенной и неотъемлемой частью деятельности профессионала, использующего любую современную компьютерную систему. Вводом, хранением, редактированием и выводом (распечатыванием в удобном виде) текста занимает часть системы, носящая название «редактор текстов». Существует множество разновидностей таких редакторов, однако ни одна из них не удовлетворяет пользователей в полной мере. Лишь малая часть потребителей вполне удовлетворена ими, и составляют эту малую часть... сами производители — программисты.

Собственно, почти все известные разновидности изначально были предназначены для удовлетворения нужд программистов, которые наивно считают, что такого рода редакторы устроят всех, часто забывая, однако, что сами они имеют дело с текстом весьма специфической структуры. Школьному же учителю литературы, например, весьма непривычно было бы оформлять свой текст так, чтобы каждая фраза начиналась с красной строки и заканчивалась точкой с запятой.

Итак, с сапогами пока лишь сапожники. Редакторы текстов, как и персональные компьютеры в целом, предназначены, в основном, для использования непрограммирующими специалистами, а они не удовлетворены современным состоянием текстообработки.

Сколько же должно быть редакторов? Ясно, что представители каждой из профессий, пользующиеся компьютерами, нуждаются в достаточно специализированном редакторе текстов. Скажем, полиграфист более всего заинтересован в развитых средствах оформления (использование различных шрифтов, выделений, включение графических элементов и т. д.), тогда как для журналиста более важной является простота работы с прозаическим текстом и легкость использования машинного архива (компоновка

фрагментов заранее подготовленных материалов).

Может показаться, что имеется необходимость создания множества разнообразных редакторов, каждый из которых рассчитан на обеспечение вполне определенной профессиональной деятельности. Понятно, что разработка этого множества обошлась бы весьма дорого. С другой стороны, предпринимаются попытки создания такого редактора, который удовлетворял бы представителей любой профессии. Многих эта идея соблазняет своей кажущейся дешевизной в производстве: им представляется, что легче и дешевле наладить серийный выпуск одного редактора, нежели многих.

Однако легко понять, что такой универсальный редактор, предназначенный для всех сразу, не устроит никого в отдельности. Даже если в подобной разработке и удалось бы учесть потребности самых разных профессионалов, изготовленная система оказалась бы настолько громоздкой и отнимала столько сил на свое освоение, что практическое использование ее было бы невозможным. На всякий чих не наздравствуешься!

Ровно столько, сколько нужно. Любой специалист, для которого компьютер — одно из вспомогательных средств в его деятельности, хотел бы иметь компактный, свденный до минимума набор команд, который позволит ему, не прилагая лишних усилий, получить от компьютера необходимую помощь. Поэтому представляется разумным избрать третий путь: составить «карту» видов профессиональной деятельности, нуждающихся в автоматизации текстообработки. Такая карта может быть создана на основе классификации текстов, обрабатываемых представителями различных профессий. Она позволит увидеть, какие виды профессиональной деятельности требуют сходных средств текстообработки, и создать несколько типовых систем, каждая из которых способна удовлетворить вполне определенный круг профессий.

Таких типовых систем потребуется не слишком много, и каждую из них можно будет превратить в конкретную (используемую во вполне определенном виде деятельности) с помощью несложных преобразований. Подобный подход позволил бы и учесть специфику различных профессий, и избавить государство от огромного количества дорого-

стоящих разработок. Надеемся, что эта задача будет в разумные сроки решена заинтересованными организациями.

## Основные виды текста и их характеристики

Конкретные способы текстообработки обусловлены разнообразием текстовых структур, с которыми приходится иметь дело в каждом конкретном случае. Выделим два аспекта взаимодействия человека с текстом — изобразительный и операционный. Изоструктура возникает у человека, когда он рассматривает какое-либо изображение текста. Операционную структуру составляют те элементы, которыми пользователь может реально манипулировать (переставлять местами, удалять и т. д.). В некоторых видах текстов мы потребуем возможно более полного соответствия между этими структурами, в то время как в других удобно, наоборот, их различать.

Перечислим некоторые виды текстов, наиболее часто встречающиеся, и укажем их отличительные признаки.

Самым распространенным является текст *прозаический*. Обычные структурные элементы его (как изобразительные, так и операционные) — знак (буква, цифра и т. п.), слово, предложение, абзац, параграф (чаще всего в деловой и учебной литературе), глава, раздел. Важнейшие структурные связи: иерархия (глава состоит из абзацев, абзацы из слов и т. п.) и следование (например, слов друг за другом).

В *поэтическом* тексте на прозаическую иерархию накладывается еще ритмическая структура, отражающая музыкальные корни стихотворения. Таким образом, добавляются новые элементы — строка и строфа.

*Табличный* текст заметно отличается от вышеназванных. Его основные элементы — строка, столбец и графа (образующаяся на пересечении строки и столбца). Текст внутри графы таблицы может обладать своей собственной структурой (например, он может быть прозаическим или таблицей), а столбцы и строки могут объединяться во взаимосвязанные группы («план», «факт», «процент») — это делает реальные таблицы весьма сложными образованиями. Объединяет их наличие двух направлений следования — по строке и по столбцу. Заметим, что предыдущие виды текста обладали лишь одним направлением следования: по главам, по предложениям, по строкам — движение как бы по одной оси, но с разным шагом.

Особый вид текстов — *математические формулы*. Их структурными элементами являются символ, индекс, степень, числитель,

знаменатель и т. д. Основные структурные связи, как ни странно, совпадают со связями в прозаическом тексте (иерархия и следование).

Переходя к совершенно особому виду текста — *графике*, сделаем следующее замечание. Поскольку любой изображаемый текст рассчитан на зрительное восприятие и «атомарные» элементы его восходят в конечном счете к пиктографии и иероглифике древнейших систем письменности, постольку и любое целостное графическое изображение (от буквы до картины) может рассматриваться как единица текста. Графическим приходится считать тот элемент, в котором (в процессе восприятия данного текста) мы не выделяем никакой структуры. С операционной точки зрения он также является неделимой единицей.

К разряду *псевдографических* отнесем все тексты, которые с изобразительной точки зрения являются графическими, а с операционной — нет, поскольку составлены из некоторых типовых элементов (пример такого текста — чертеж).

Распространенным видом псевдографического текста является мозаичный. Его элементы размещены на одинаковых геометрических фигурках (квадратах, шестиугольниках и т. п.), которые, стыкуясь друг с другом, образуют различные картинки. Такая фигурка является единственным операционным элементом данного текста.

*Шаблонный* текст встречается весьма часто в виде анкет, заявлений и других трафаретных документов. С изобразительной точки зрения это, как правило, прозаический текст, он состоит из двух частей — постоянной и переменной. Содержание и размещение переменной части строго определено. Единственным операционным элементом текстов данного вида является поле (единица переменной части).

Прежде чем перейти к такому виду текстов, как *компьютерные программы*, опишем еще одну специфическую структуру — *цепочку знаков*. Любой отрезок текста такого рода — это либо знак, либо цепочка знаков. От прозы рассматриваемый текст отличается примитивностью своей структуры. Если проза обычно строится по иерархическому принципу (глава — абзац — предложение и т. д.), то цепочка знаков совершенно однородна.

Компьютерная программа (как вид текста) представляет собой последовательный набор строк, каждая из которых является цепочкой знаков. Так как несколько таких цепочек составляют новую, программисты часто рассматривают всю программу как единую цепочку знаков. Отсюда понятно, почему

большинство редакторов ориентированы на тексты, основными операционными элементами которых как раз и являются строки и цепочки знаков.

Мы привели примеры достаточно распространенных видов текста. Часто, однако, мы сталкиваемся со смешанными текстами, как бы композициями из текстов разных видов (например, страница прозы с картинкой посередине или таблица, содержащая в своих графах прозаический текст).

Достаточно часто, к сожалению, встречается в редакторах неоправданное несоответствие между изобразительной и операционной структурами. Например, человек не замечает на изобразительном уровне сколько-нибудь существенной разницы между двумя структурными элементами, а оказывается, что один из них является операционным (с ним можно что-то сделать), а другой — нет. Это ситуация операционной неоднородности. Ее можно наблюдать в большинстве табличных редакторов, где легко удаляется или вставляется строка, в то время как со столбцом ничего такого сделать нельзя. Бывает, что изобразительная структура вообще не имеет операционной поддержки. Скажем, табличная строка нередко состоит из нескольких экранных, а операции позволяют нам работать только с экранными строками.

## Редактирование прозаического текста

Рассмотрим подробнее работу с наиболее употребительным текстом — прозаическим. Здесь мы предложим набросок редактора, содержащий среди прочего идеи, уже реализованные в таких системах, как ДИС, СМОЛ-ЛТОК-80. В целом же он еще ждет своего воплощения.

**Обстановка.** Пользователь видит перед собой экран; перед экраном на столе несколько клавиатур — обычная, напоминающая клавиатуру пишущей машинки, и так называемые функциональные (состоящие из 10—15 клавиш). Некоторые компьютеры снабжены также устройством, позволяющим перемещать указатель текста (курсор) на экране (таких устройств у нас в стране, к сожалению, еще крайне мало).

Экран дает возможность видеть интересующий пользователя в данный момент фрагмент текста. Обычная клавиатура позволяет набрать, а затем дополнить сам текст. Остальные компоненты служат для фиксации и исправления определенного места текста.

**Ввод.** Прежде всего остановимся на вопросах, связанных с достаточно удобным набором текста (его вводом в систему). В на-

стоящее время текст принято вводить в виде цепочки знаков (эта работа, как правило, осуществляется специально подготовленным оператором), а затем «размечать» введенный текст (отчасти это делается автоматически, остальное осуществляет технический редактор).

Разметка в отношении прозаического текста заключается в членении его на абзацы, отделы, главы и т. д. Однако этот метод работы неудобен, когда над текстом работает всего один человек (например, педагог или журналист). В таком случае предпочтительнее совершать разметку текста по ходу его набора при помощи специальных клавиш на функциональной клавиатуре. Например, начиная набор очередной главы, пользователь нажимает клавишу «глава» (после чего соответствующая надпись появляется на экране в середине новой строки) и к ней прибавляет номер, набранный на обычной клавиатуре. Подобная же операция совершается и при переходе к новому абзацу: нажимается «абзац» и т. д. Таким образом, набранный текст уже оказывается размеченным.

**Поиск.** Самым сложным этапом компьютерной работы с текстом является его редактирование. Первое требование, предъявляемое к компьютеру на этом этапе работы, — удобное и быстрое нахождение определенного места в тексте. Существует много способов вывести на экран требуемый фрагмент. Опшем наиболее удачные.

Первый предполагает условное разделение всего текста на фрагменты, каждый из которых целиком помещается на экране. При таком способе нужный фрагмент находится путем «перелистывания», которое совершается с помощью двух клавиш, позволяющих «передвигать» текст в направлении как к его концу, так и к началу. Второй способ, позволяющий плавно передвигать текст (не разделяя его заранее на фрагменты), характеризуется специальным термином «скролл» (свиток). Если компьютер не снабжен специальным устройством типа «мышь», то приходится выделить две клавиши, с помощью которых задается направление плавного (с фиксированной скоростью) движения текста. Устройство типа «мышь» позволяет не только задавать направление, но и регулировать скорость прохождения текста по экрану. Это дает возможность быстро пропускать места текста, не интересующие пользователя в данный момент, и подробно просматривать нужные фрагменты.

Третий способ поиска заключается в передвижении по структурным элементам текста. Типичная команда в этом случае: «перейти к следующему (предыдущему) предложению».

нию (абзацу)». Ее удобно давать, нажимая последовательно на клавишу «перейти» (к последующему, предыдущему) и на клавишу функциональной клавиатуры, обозначающую требуемый структурный элемент (главу, абзац и т. д.).

Структурные элементы текста могут нумероваться, что облегчит поиск: «перейти к девятой главе, четвертому параграфу». Эта процедура потребует последовательного нажатия клавиш «перейти», «9», «глава», «4», «параграф».

Четвертый способ состоит в предварительной разметке текста с выделением мест, требующих исправления («маршрутизация»). Он позволяет выстраивать по цепочке иерархически неоднородные элементы текста (главу, абзац, предложение и т. д.), находящиеся совсем в разных местах. Это удобно при работе, требующей неоднократного возвращения к местам, уже просмотренным ранее. Такой «набор закладок» отличается от книжного тем, что отмечается не разворот, а любой элемент текста.

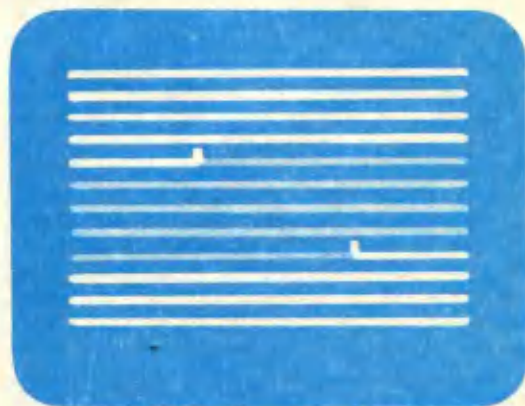
Здесь требуется введение дополнительных клавиш: «вперед по маршруту», «назад по маршруту», «вложить закладку», «вытащить закладку».

Упомянем еще один — пятый — способ работы, который называют обычно «поиск по образцу». Фрагмент, включающий в себя данный образец (слово, предложение или цепочку знаков), требуется вывести на экран. Для этого образец при помощи клавиатуры набирается на специально выделенном на экране поле. Затем пользователь нажимает клавишу «поиск», и нужный фрагмент оказывается перед ним. При повторении команды «поиск» на экране появляются следующие фрагменты, содержащие тот же образец.

Работа с текстом. Познакомившись с несколькими способами нахождения нужного для работы места, перейдем непосредственно к основным командам редактирования. Три главные команды — «вставить», «удалить» и «заменить». В некоторых редакторах команда «заменить» отсутствует, ее функцию выполняет последовательное выполнение команд «удалить» и «вставить».

При нажатии клавиши «вставить» на экране отводится свободное пространство для набора необходимого фрагмента. Набор завершается нажатием клавиши «выбор», затем курсором мы указываем место, начиная с которого фрагмент должен быть включен в текст. Еще раз нажимается клавиша «выбор», с помощью которой, как правило, осуществляется завершение операции: на экране появляется уже отредактированный текст.

После нажатия клавиши «удалить» нужно



указать курсором на экране начало подлежащего удалению фрагмента и нажать клавишу «выбор». Переместив курсор и указав конец удаляемого фрагмента, мы вторично нажимаем «выбор». После третьего его нажатия на экране возникает исправленный текст. Третье нажатие может показаться лишним, однако такой вариант открывает некоторые возможности, обсуждение которых выходит, к сожалению, за рамки статьи.

Последовательность действий после нажатия клавиши «заменить» такая же, как при вставке, с тем отличием, что после указания курсором и клавишей «выбор» места, начиная с которого новый фрагмент должен быть включен в текст, мы должны указать конец той части текста, которую заменяет фрагмент. Это делается как при удалении.

При машинном редактировании зачастую применяется так называемый конверт. Он используется, например, для перенесения некоего фрагмента из одного места текста в другое. Для этого переносимый фрагмент удаляется и «кладется в конверт» (клавиша «конверт» нажимается непосредственно после определения удаляемого фрагмента). При введении же переносимого фрагмента в новый



контекст сразу после команды «вставить» нажимается клавиша «конверт», и фрагмент не приходится набирать: он как бы вынимается из «конверта». Этот способ позволяет значительно облегчить редактирование текста, например, путем отбора в «конверт» сразу нескольких фрагментов, сводимых там воедино и помещаемых в определенное место текста.

Для дублирования текста используется клавиша «копия». При этом на экране описанным выше способом (с помощью курсора) выделяют подлежащий дублированию фрагмент, затем пользователь находит нужное место (выводит его на экран) и отмечает с помощью курсора, куда именно требуется ввести дубликат. Операция, как всегда, завершается нажатием клавиши «выбор». Дубликат можно также поместить в «конверт», что оказывается удобным для последующего включения его в несколько мест. В «конверт» при необходимости можно ввести также произвольный фрагмент, не встречавшийся до этого в тексте.

Отменить выполнение любой команды можно клавишей «сброс».

Для ускорения редактирования удобно иметь возможность окончательно оформлен-

ные его части как бы «выносить за кадр», чтобы не тратить время на многократное «пролистывание». Для этого можно использовать клавишу «спрятать», после нажатия которой заранее отмеченная часть текста временно станет невидимой. Так, скажем, полностью закончив работу над пятой главой текста, мы объявляем ее «спрятанной», после чего при повторном «пролистывании» текста на экране появляется только ее заголовок. Когда же текст, «спрятанный» описанным выше способом, вновь требуется сделать видимым, мы нажимаем клавишу «обнаружить», и нужная часть текста появляется на экране.

Целый ряд удобных операций позволяет осуществить средство, называемое условно словарем. В этот словарь уже при вводе текста в машину автоматически вносятся и располагаются в алфавитном порядке все присутствующие в тексте слова. Такой словарь можно использовать при корректуре: просмотрев все встречающиеся слова и их формы, убедиться в правильности их написания. Особенно облегчит словарь корректуру текстов значительного объема, так как в этом случае объем словаря будет значительно

59



меньше объема текста. Словарь дает также возможность найти все случаи употребления данного слова или определенной его формы и, таким образом, может быть использован для нахождения определенных участков по ключевым словам, для поиска параллельных мест и т. д. В словарь пользователь может также ввести интересующие его словосочетания, фразеологизмы и т. д., что позволит быстро отыскивать фрагменты текста, содержащие эти словосочетания. Словарь может также ускорить операции типа замены какого-либо слова его синонимом по всему тексту: для этого достаточно произвести такую замену в самом словаре.

**Изображение.** Выше мы говорили о средствах, помогающих в работе над семантикой и грамматикой текста. Теперь перечислим те средства, которые облегчают работу над «внешним видом» при непосредственной подготовке его к печати. Здесь важно задать формат страницы, гарнитуру, выделения (курсив, жирность, подчеркивание и др.), прочие собственно полиграфические характеристики.

Упомянем в первую очередь такое важное средство, как описатель. Каждой структурной единице текста соответствует свой описатель (есть описатели абзаца, главы, всего текста и др.). В описатель всего текста заносятся такие характеристики, как язык, на котором он написан (это важно для выбора формы литер и алгоритма переноса), формат полосы и расположение ее на странице, межстрочные интервалы и др. В описатель главы или абзаца заносятся характеристики этого структурного элемента (например, для абзаца — размер отступа). Описатель каждого структурного элемента текста в обязательном порядке включает в себя характеристику шрифта, что позволяет выделить особым шрифтом любой фрагмент (если такая возможность предоставляется печатающим устройством), а также выделения (курсив, разрядку и др.), присутствующие, скажем, в предложении или абзаце.

Важно, что так можно не только описать все однотипные структурные единицы данного текста разом, но и оговорить индивидуальные особенности некоторых из них (например, охарактеризовать места, требующие специального шрифта или выделяемые курсивом).

**Верстка.** Из сказанного выше следует, что работа над текстом перед его распечатыванием как раз и состоит во вдумчивой и аккуратной подготовке описателей.

После завершения этой подготовительной работы подается команда «верстка», по которой машина создает текст, готовый к печати.

Перед печатью, однако, пользователь дол-

жен иметь возможность еще раз просмотреть целиком сверстанный текст. При этом особое внимание следует уделить качеству машинного переноса, который в сложных случаях может отклониться от нормы. Желательно, чтобы алгоритм переноса сам обнаруживал такие места и запрашивал пользователя о выборе правильного варианта. Просмотрев текст и при необходимости внеся исправления, пользователь снова подает команду «верстка».

Окончательный вариант подготовленного таким образом текста подается на имеющееся при компьютере устройство печати. Спектр этих устройств весьма широк — от принтера, воспринимающего лишь прописные буквы, до фотонаборного автомата, выдающего гранки текста в полиграфическом исполнении, и лазерного принтера, в короткий срок печатающего тысячи цветных экземпляров.

## Дальнейшее структурирование

**Текст как предмет.** У читателя мог возникнуть ряд вопросов, на которые мы попытаемся здесь ответить. Для этого нужно зафиксировать некоторые понятия. Условимся различать в тексте три уровня — смысловой, знаковый (языковый) и уровень носителя. Чаще всего, говоря «текст», подразумевают его знаково-смысловой слой. При таком понимании соотношение между текстом и его носителем можно наглядно представить себе так: если текст мы метафорически соотнесем с душой, то носитель текста будет выглядеть телом, в котором воплощена душа — текст. Так, одно и то же может быть записано и в обычного вида книге (кодексе), и в кожаном свитке, и на каменной стеле, а также на экране дисплея и магнитном диске.

Понятно, что каждый носитель текста обладает своей собственной структурой (соотношением элементов). Например, книга составлена из листов, а на экране алфавитно-цифрового дисплея зафиксированы знакоместа. Структура самого текста входит в сложные взаимоотношения со структурой носителя. Бывают случаи, когда структура текста вынуждает радикально изменить структуру носителя. Так, например, на рубеже нашей эры, в связи с более широким распространением грамотности и народившейся потребностью в прямых ссылках и цитатах (полемика раннехристианских авторов с язычниками), кожаный или папирусный свиток постепенно заменяется всем нам знакомым кодексом. Это было обусловлено тем, что многие наиболее употребительные тексты (библейские и классические) были раз-

биты на небольшие пронумерованные фрагменты, которые необходимо было быстро находить и сопоставлять между собой, что значительно легче осуществляется перелистыванием страниц, чем прокручиванием объемистого свитка.

Чаще, однако, встречаются случаи обратного воздействия — структуры носителя на структуру текста. Это, как правило, связано с некоторыми ограничениями в самом изображении текста. Так, например, разметка таблиц учитывает предполагаемые размеры книжных страниц.

Рассматривая различные тексты, мы для простоты не проводили четкого различия между структурой самого текста и структурой носителя, однако между ними возможны самые разные взаимоотношения. Так, например, и проза, и математические формулы организованы иерархически, но отображаются на носителе они очень по-разному.

**«Живой» носитель.** Редактируя текст при помощи ЭВМ, мы непосредственно работаем с некоторым носителем текста, изображенным на экране, причем обладающим замечательным свойством: мы можем быть уверены в том, что операция, произведенная над этим носителем, фактически является операцией над самим текстом.

Так, например, при редактировании математических формул на алфавитно-цифровом дисплее формула «А умножить на В в квадрате и разделить на С» будет изображена в виде  $A * B^{**2} / C$ . Для того чтобы изменить степень В с квадрата на куб, надо в формуле заменить цифру 2 на цифру 3. При этом мы должны быть уверены, что сделанное нами на экране исправление является исправлением «настоящей», первоначально заданной формулы. Именно эта «настоящая» формула и появится при выводе на печатающее устройство.

**Разделяй и властвуй.** Из сказанного ясно, что полезно иметь два набора управляющих команд: один из них должен быть ориентирован на структуру носителя, другой — на структуру данного текста. Команды каждого из этих наборов можно разделить на небольшое количество часто используемых команд и большее число употребляемых заметно реже. Команды, используемые часто, удобно подавать с функциональной клавиатуры, а остальные при необходимости выбирать из так называемого меню на экране.

Очень полезно на клавиатуре, ориентированной на структуру текста, иметь пару клавиш, которую мы назовем «масштаб». Они будут указывать на «масштаб» выделяемого курсором (и вследствие этого иначе отображаемого на экране) элемента текста, на иерархический уровень, с элементами которого

предстоит работа (знак, слово, предложение, абзац, глава и др.). Например, если необходимо перегруппировать несколько абзацев, нам удобно установить масштаб «абзац»; тогда в дальнейшем не придется указывать, какие элементы подлежат перегруппировке. Две клавиши нужны для того, чтобы увеличивать масштаб или уменьшать его.

Установив масштаб, мы перестанем нуждаться в том, чтобы в команде «перейти к следующему» всякий раз заново указывать соответствующий структурный элемент (он уже задан).

Во второй же клавиатуре, относящейся к работе с носителем текста, клавиши, применяемые аналогичным образом, выполняют как бы функцию увеличительных или уменьшительных стекол, позволяя то уместить на экране крупный фрагмент текста (без детализации), то перейти к показу деталей крупным планом. Это должно способствовать наиболее удобному размещению отдельных фрагментов текста на экране.

Не перечисляя подробно всех команд движения на двух упомянутых клавиатурах, укажем характерные отличия этих команд. Если для клавиатуры, ориентированной на текст, характерны команды типа «перейти к следующему предложению» или «выйти на главу 5», то для клавиатуры, ориентированной на носитель, характерны команды типа «встать в начало следующей строки» или «перейти к следующему кадру».

Одним из удобных приемов текстообработки является разбиение экрана на «окна», т. е. на некоторые фиксированные прямоугольники. С подобным случаем мы уже встречались при описании команд «вставить», «заменить». «Конверт» также изображается на экране в виде «окна». Часто встречаются виды работы, требующие одновременного использования нескольких текстов; например, из трех чужих статей «научный работник» komponует свою докторскую диссертацию. Для этого ему понадобятся на экране четыре «окна»: три — для чужих текстов и одно, вначале пустое, — для своего, вновь создаваемого. При такой работе к числу команд требуется прибавить еще одну: «перейти к окну № ...».

В каждом виде работ с текстами можно выделить небольшой компактный набор необходимых операций. Однако, вместо того чтобы внимательно изучить такие наборы для каждого конкретного вида работ, многие создатели редакторов предпочитают решить этот вопрос введением так называемых макрокоманд (макрокоманда — зафиксированная последовательность некоторых операций данного редактора, которую можно испол-

зывать каждый раз как единое целое).

Пользователю, по сути, предлагают набор разрозненных элементарных операций и возможность конструирования из них макрокоманд, при этом набор элементарных операций не связывают с выполнением именно требуемого вида работы, а пытаются, напротив, сделать универсальным, пригодным для конструирования из него макрокоманд на все случаи жизни. Таким образом, пользователю навязывается не свойственная ему фактически программистская деятельность. Кроме того этот «универсальный» набор всегда оказывается неполным, и часть необходимых пользователю операций осуществить трудно или невозможно.

В качестве альтернативы такому подходу можно указать на ДИС (диалоговую издательскую систему), разработанную Л. Ратнером с сотрудниками в ленинградском филиале ГипроНИИполиграфа. В ней каждому виду работ (вводу, корректуре, научному и техническому редактированию и др.) соответствует набор удобно выбранных операций.

Однако даже если набор операций достаточно удобен, в процессе длительной работы может возникнуть потребность во введении новых операций. Этим должны заниматься не потребители конкретных систем редактирования, а специалисты, работающие с типовыми системами.

**«Жизнь» текстов.** Представляется важной также проблема сохранения предыдущих версий текста. Как правило, очередная работа с текстом проводится над последней его версией, однако в ряде случаев оказывается полезным привлечь и предыдущие. Пользователь может по собственному усмотрению счесть тот или иной вариант текста его очередной версией и присвоить этому варианту номер (который может задаваться автоматически) или наименование. Часто этот вопрос решают следующим образом. Считается, что

текст при всяком новом входе его в редактор автоматически становится своей очередной версией. Это плохо: моменты входа в редактор и выхода из него задаются временем, к которому они приурочены, а версию текста удобнее связывать с некоторым пластом изменений его содержания (понятно, что такой подход к тексту способен осуществлять только сам человек).

Здесь мы вплотную подходим к проблеме обратимости текста. Необходимо иметь возможность возвращаться на одну операцию назад, т. е. к предыдущей редакции: это исключает возможность внесения в текст непоправимых ошибок. Очень полезно и иметь возможность вернуться к любой из прежних редакций текста, т. е. «прокрутить» его «во времени» на любое число операций назад. Все это имеет смысл осуществлять в рамках именно данной, текущей версии. Естественно, все предыдущие версии тоже должны сохраняться.

В редакторе должны быть реализованы также удобные механизмы объединения нескольких текстов в новый текст (соответственно, с новым названием) и разделения существующего текста на несколько новых. Реализация «возврата по версиям» позволит нам легко восстановить историю создания текста: возвращаясь назад, к предыдущим версиям, мы получим возможность проследить фазы работы над текстом и выяснить как его первоначальные источники, так и привлеченные в ходе работы.

Конечно, все эти предложения не догма уже хотя бы потому, что особые редакторы нужны не только для текстов разных типов, но и для пользователей с разными характерами. Будем надеяться, что уже скоро программист, эргономист и психолог сядут за один стол с будущим пользователем и начнут решать эту проблему.

*Дорогие читатели! Из наших публикаций вы уже знаете о некоторых возможностях Таллинского научно-учебного центра по поставке программного обеспечения. Если вы хотите получить полные сведения об имеющихся в его ассортименте программах, вы можете встать на информационное обслуживание. Для этого следует направить в адрес центра (см.: Информатика и образование. 1988. № 2) гарантийное письмо с соответствующей просьбой и обязательством об оплате (информационное обслуживание стоит 114 рублей в год).*



В. АНИСИМОВ,  
НИИ школ Минпроса РСФСР

Л. СОЛОВЬЕВ,  
ЦИУУ Минпроса РСФСР

## Учебный пакет прикладных программ «МОТОР»

Для широкого круга пользователей-непрограммистов общение с ЭВМ сводится к работе с готовыми программными средствами в виде пакетов прикладных программ. Применение в школе учебных пакетов прикладных программ для решения задач на ЭВМ позволяет показать учащимся особенности машинно-ориентированного подхода к поиску способа решения, привить навыки работы с пакетами и библиотеками программ.

От большинства пользователей ЭВМ требуется не только умение конструировать новые алгоритмы, но и использовать все многообразие готовых программ, работа с которыми уже сегодня главным образом определяет характер общения человека с вычислительной техникой. Привлекая для решения задач готовые программы, пользователь наделяет ЭВМ как исполнителя алгоритмов новыми качествами. Строя алгоритм, основанный на работе программ пакета, человек распоряжается более мощными возможностями ЭВМ по сравнению с теми, которые предоставляют ему команды языка программирования. Готовые программы дают учащимся реальную возможность еще до овладения техникой программирования справляться за время урока с такими задачами, которые не решаются без ЭВМ или требуют слишком больших усилий.

Работа с пакетом «МОТОР» моделирует деятельность реального пользователя по решению задачи на ЭВМ средствами любого другого пакета прикладных программ. Она охватывает все этапы — от составления алгоритма до документирования результатов. Работа с «МОТОРОм» не требует от учащегося умений составлять программы на языке программирования и может проводиться на начальном этапе общения с ЭВМ.

Пакет прикладных программ «МОТОР» предназначен для решения класса задач, сводимых к исследованию свойств функций одной переменной, которые должны быть заданы в виде формул. Функции могут зависеть от параметров. Обязательным условием является однозначность и непрерывность функций на заданном отрезке.

Многие из таких задач решаются методом математического моделирования, когда необ-

ходимо найти математическое описание процесса, явления, ситуации, а затем исследовать свойства функции. Изучению функциональных зависимостей в школьном курсе математики придается большое значение. Однако трудоемкость решения таких задач классическими безмашинными методами в значительной мере зависит от вида исследуемых функций, и многие задачи не могут быть предложены учащимся. Особенно это касается примеров, взятых из практики, когда требуется получение результатов в численном виде. Использование компьютеров с пакетом прикладных программ «МОТОР» расширяет класс задач, с которыми школьники могут справиться самостоятельно, и приближает их к диалогу с ЭВМ, поиску рациональных способов решения задачи, знакомит с возможностями и ограничениями численных методов исследования функций.

### Состав программ пакета «МОТОР»

В состав пакета прикладных программ (ППП) «МОТОР» входят программы-функции и программы обработки. Кроме того, в «МОТОРе» есть средства управления, взаимодействия с учащимся и контроля работы программ.

Программа-функция представляет собой средства задания исследуемой функции  $y=f(x)$ , в которые входят: выражение функции, определение ее параметров и описание кадра с изображением математической формулы  $f(x)$ . Выражение функции записывается по правилам языка программирования в форме, доступной для обращения из программ обработки. Это выражение может иметь от одного до четырех параметров, значения которых задаются в диалоговом режиме.

Содержание программы-функции определяется исследуемой математической моделью, которая может быть задана в условии задачи или получена учащимися самостоятельно. Авторами подготовлен набор из восьми заменяемых программ-функций, которые можно по выбору включать в состав пакета. При необходимости учитель или ученик могут

легко перенастроить программу-функцию под конкретную задачу.

Программы обработки обеспечивают исследование функций (без вычисления производных) и нахождение определенного интеграла. Учащиеся, используя три формы представления функции (в виде формулы, графика и таблицы), могут найти наименьшее и наибольшее значение функции, определить, в каких точках она пересекает ось  $OX$ , вычислить площадь криволинейной трапеции, определить участки, где функция возрастает, убывает или остается постоянной, проверить периодичность функции и установить, является она четной или нечетной.

Каждая из программ обработки реализует один из алгоритмов, позволяя получить либо какую-нибудь характеристику функции, либо представить функцию в удобном для пользователя виде. Параметрами настройки программ обработки являются границы отрезка на оси  $OX$  (а при построении графиков и на оси  $OY$ ), на котором рассматривается исследуемая функция.

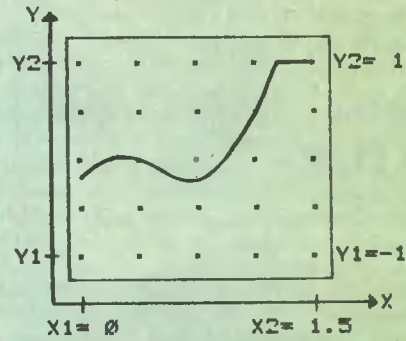
В состав пакета «МОТОР» включены шесть программ обработки:

1) Программа «РАМКА» дает таблицу значений функции  $y=f(x)$  (рис. 1).

$X_1, X_2$  — левая и правая границы отрезка, на котором вычисляются значения функции. Программа делит заданный отрезок на 8 равных частей, вычисляет и выводит на экран дисплея 9 значений функции, включая ее значения в граничных точках.

2) Программа «ОКНО» строит график функции  $y=f(x)$  (рис. 2).  $X_1, X_2$  — границы отрезка из области определения функции, а  $Y_1, Y_2$  — границы области значений, в которых представляет интерес поведение функции. Параметры настройки определяют на плоскости  $XOY$  прямоугольную область, в которой будет представлен график. Программа выводит на экран изображение прямо-

МОТОР: работает программа ОКНО



Изменить  $X_1, X_2, Y_1, Y_2$ ? Да(1), Нет(0)

угольной области и располагающуюся в ней часть графика. Если отрезок  $[Y_1; Y_2]$  не покрывает весь диапазон  $[f(X_1); f(X_2)]$ , то выходящая за эту прямоугольную область часть графика заменяется отрезком прямой на ее границе. Программа делит заданный отрезок на 30 частей и вычисляет значения  $f(x)$  в полученных точках, а затем выдает на экран кусочно-линейное изображение функции.

3) Программа «ТРЕК» находит наибольшее и наименьшее значения функции и значения аргумента, при которых они достигаются (рис. 3).

$X_1, X_2$  — границы отрезка, на котором отыскивается максимум и минимум. Программа делит заданный отрезок на 50 равных частей и вычисляет в полученных точках значения функции. Из этого набора значений программа выбирает максимальное и минимальное. Кроме того, программа сообщает о погрешности. В качестве погрешности указывается величина, составляющая 0,02 длины заданного отрезка.

4) Программа «КРОТ» находит нуль

МОТОР: работает программа РАМКА  
 $X_1=0$        $X_2=1.5$

X	Y(X)
$X_1=0$	-0.2000
.1875	0.0373
.375	0.0109
.5625	-0.1209
.75	-0.2000
.9375	-0.0682
1.125	0.4328
1.3125	1.4611
$X_2=1.5$	3.1750

Изменить  $X_1, X_2$ ? [Да(1), Нет(0)]

МОТОР: работает программа ТРЕК  
 $X_1=0$        $X_2=0.98$

■ НАИБОЛЬШЕЕ значение  
функции  $Y(X)$  близко к 0.050  
Оно достигается  
в точке  $X=2.35E-01$

■ НАИМЕНЬШЕЕ значение  
функции  $Y(X)$  близко к -0.200  
Оно достигается  
в точке  $X=0.00E+00$

■ ОШИБКА  $\Delta X$  не более 0.020

Изменить  $X_1, X_2$ ? [Да(1), Нет(0)]

МОТОР: работает программа КРОТ

X1= 0 X2= .255

■ При X= 1.3012E-01

Y(X) обращается практически  
в нуль!

■ Ошибка Δx не более ± 1.E-04

Изменить X1, X2? [Да(1), Нет(0)]

функции  $y=f(x)$ , т. е. вещественный корень уравнения  $f(x)=0$  (рис. 4).

X1, X2 — границы отрезка, на котором предполагается наличие корня. Если на концах заданного отрезка  $f(x)$  имеет значения разных знаков, программа находит нуль этой функции и сообщает, с какой ошибкой вычислено значение нуля. В том случае, если на заданном отрезке нет нуля либо на концах отрезка функция имеет значения одного знака, на экран выводится специальное сообщение. Программа использует метод половинного деления, который применяется последовательно 10 раз. В результате выполнения таких действий длина отрезка, содержащего нуль функции, сокращается приблизительно в 1000 раз. Эта длина и составляет ошибку определения нуля.

5) Программа «РЕКА» находит участки убывания, возрастания и постоянства функции (рис. 5).

X1, X2 — границы отрезка, на котором исследуется монотонность функции. Программа выдает список участков отрезка [X1; X2] с указанием границ каждого участка и

МОТОР: работает программа РЕКА

X1= 0 X2=1.5

■ На отрезке от 0  
до .255 Y'(X) > 0

■ На отрезке от .255  
до .75 Y'(X) < 0

■ На отрезке от .75  
до 1.5 Y'(X) > 0

■ Ошибка ΔX меньше 0.02

Изменить X1, X2? [Да(1), Нет(0)]

МОТОР: работает программа МОРЕ

X1= 1 X2=1.5

■ Вычислен интеграл

$$\int_1^{1.5} Y(x) dx = 6.189E-01$$

■ Число трапеций 32

■ Ошибка меньше 0.02

Изменить X1, X2? [Да(1), Нет(0)]

знака на нем первой производной  $y'(x)$ , сообщает о погрешности. Программа делит отрезок на 50 равных частей, находит знаки приращений функции для каждой из них. Далее смежные части с одинаковыми знаками приращений объединяются в непрерывные участки. Погрешность определения границ между участками не превышает 2% от длины отрезка.

6) Программа «МОРЕ» вычисляет определенный интеграл (рис. 6).

X1, X2 — нижний и верхний пределы интегрирования. Программа выдает математический символ и значение интеграла, сообщает число трапеций и погрешность вычислений. В программе используется метод трапеций. Число разбиений отрезка [X1; X2] выбирается автоматически в зависимости от вида функции. Максимальное число трапеций равно 256, минимальное — 2. В качестве погрешности программа сообщает абсолютную величину разности двух последних сумм.

Все программы составлены на языке Бейсик. Пакет программ может быть адаптирован на различные персональные компьютеры. Первая версия была реализована на ЭВМ СМ-4 с операционной системой ОС РВ. Подготовлена вторая версия «МОТОРа», рассчитанная на компьютеры «Ямаха», и третья версия — для КУВТ «Ямаха» и «Корвет». На основе разработанной методики одним из авторов были организованы занятия и проведен открытый урок с пакетом «МОТОР» с применением восьми ПЭВМ «Агат» в школе № 502 Москвы.

Кроме персональных компьютеров, для реализации пакета могут быть использованы программируемые микрокалькуляторы. При этом, естественно, учащиеся лишаются возможности графического представления функции, а анализ таблицы значений требует занесения их в тетрадь. Однако это не лишает микрокалькуляторный пакет программ функ-

циональной полноты и раскрывает особые возможности в организации работы учащихся на уроке. Например, такие трудности, как нехватка ПМК и необходимость ввода программ в память ПМК перед началом занятия, могут быть преодолены за счет закрепления одной программы обработки за одним микрокалькулятором. Тогда в классе, имеющем по крайней мере шесть ПМК, образуется пакет прикладных программ коллективного пользования, на котором может быть построен урок по решению задач бригадами учащихся.

### Особенности работы учащихся с пакетом прикладных программ

66

Опыт работы показывает, что характер деятельности обучаемого по решению задачи на ЭВМ с применением пакета программ отличается от традиционного подхода, связанного с построением алгоритма и составлением программы для ЭВМ.

Во-первых, изменяются цели деятельности учащихся и возможности компьютера как исполнителя алгоритма решения данной задачи. Пакет предоставляет ученику возможность строить алгоритм из таких шагов, для выполнения которых ЭВМ использует целую программу. Результатом деятельности программиста является отлаженная программа, а пользователя пакета — решенная с помощью компьютера задача.

Во-вторых, изменяется содержание взаимодействия человека и ЭВМ. При работе с пакетом программ пользователь выбирает план действий, основанный на возможностях имеющихся программ и на особенностях решаемой задачи, и реализует этот план, подчиняя ему как действия ЭВМ, так и свои собственные. Ученик сам активно формирует ход решения задачи, являясь соисполнителем человеко-машинного алгоритма решения поставленной задачи. При этом он получает возможность сосредоточиться на идее решения, оставив машине рутинные вычисления.

Процесс решения задачи разделяется на ряд шагов. На каждом шаге ученик решает, какие изменения нужно внести в ход вычислений, чтобы быстрее получить результаты. Под одним шагом подразумевается либо действие учащегося, либо действие компьютера.

Ученик ведет диалог с пакетом программ. Чтобы сделать свой шаг, он сравнивает полученные результаты с требованиями условия задачи, а затем либо продолжает работу, либо принимает решение о ее окончании.

ЭВМ как исполнитель алгоритмов, используя возможности входящих в пакет прикладных программ, автоматически выполняет те

или иные операции над заданной функцией и представляет результаты в удобном для учащегося виде.

Действия ученика и ЭВМ, реализующие шаги алгоритма, чередуются, причем первый и все нечетные шаги делает школьник, а остальные — компьютер. На начальном, нулевом, шаге производится настройка программы-функции заданием требуемых значений параметров.

Выполняемые шаги могут циклически повторяться, в зависимости от промежуточных результатов. Учащийся может на любом шаге изменить вид исследуемой функции, значения параметров, выбрать другую программу обработки или задать новые значения параметров настройки, оставляя неизменной функцию.

За основу действий школьник берет выполнение программ обработки, к которым он обращается как к вспомогательным алгоритмам, и план решения задачи. План сначала может быть определен не полностью и уточняться по мере получения и оценки промежуточных результатов на определенном шаге. Такой ход решения задачи делает работу с «МОТОРОМ» творческой. Усилия человека направлены прежде всего на содержание задачи, ее математический смысл.

Работа с любым пакетом прикладных программ («МОТОР» дает учителю возможность показать учащимся эту особенность) означает использование для решения задачи уже написанных и отлаженных программ, составленных алгоритмов, проверенных на практике методов — словом, знаний и опыта, накопленного другими специалистами, решавшими такую или похожую задачу. Методическая ценность подобного опыта для учащихся заключается в том, что алгоритм воспринимается ими не как объект анализа или синтеза, а как инструмент решения поставленной задачи.

### Пример решения задачи

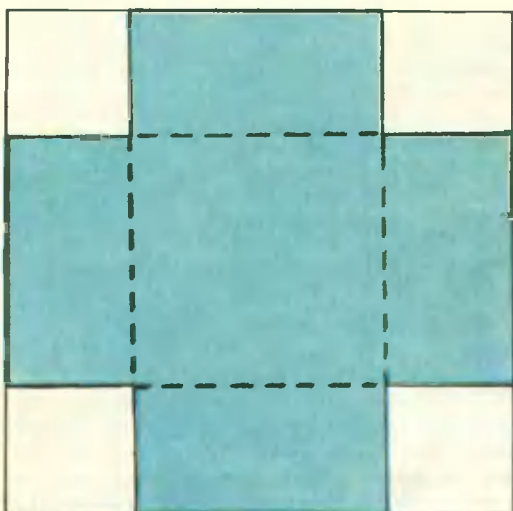
Рассмотрим методику использования пакета «МОТОР» на примере.

**Задача.** Из квадратного куска листового металла требуется выкроить развертку для изготовления короба (рис. 7). Его можно изготовить так: сделать по углам квадратные вырезы (рис. 7,а), отогнуть боковины по пунктирным линиям (рис. 7,б) и соединить боковые швы сваркой (рис. 7,в).

Определить размер квадратных вырезов, если размер заготовки  $1,5 \times 1,5$  м, а короб должен иметь объем 200 л.

Решение. Обозначим длину стороны вырезаемого квадрата через  $x$ .

1) Выразим объем короба через переменную  $x$ . Длина стороны квадратного дна короба

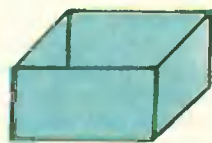


a)

x



б)



7

равна  $(1,5-2x)$  м, площадь дна составляет  $(1,5-2x)^2$  м<sup>2</sup>, а объем —  $x(1,5-2x)^2$  м<sup>3</sup>.

2) По условию, этот объем должен быть равен 200 л, или 0,2 м<sup>3</sup>. Получаем уравнение:  
 $x(1,5-2x)^2=0,2$ .

3) Раскрывая скобки, получаем уравнение третьей степени:

$$4x^3-6x^2+2,25x-0,2=0$$

4) Поскольку  $x$  выражен в метрах, зададим допустимую погрешность его определения равной 1 см, или 0,01.

Задача может быть сформулирована так: необходимо найти действительные корни этого уравнения с точностью 0,01. Задача сводится к поиску нулей функции — полинома 3-й степени  $P(x)$ :

$P(x)=Ax^3+Bx^2+Cx+D$ . Значения параметров:  $A=4$ ;  $B=-6$ ;  $C=2,25$ ;  $D=-0,2$ .

Для поиска нулей  $P(x)$  используем пакет программ «МОТОР». Составим алгоритм, по которому можно решить данное уравнение с помощью программ пакета. Инструкции, составляющие данный алгоритм, не являются формальными в смысле набора команд некоторого исполнителя, а лишь выражают цель выполняемого действия.

**алг** Решение уравнения  $Ax^3+Bx^2+Cx+D=0$  с помощью пакета «МОТОР»:

**арг** A, B, C, D;

**рез** XK1, XK2, XK3;

**нач** Задать в программу-функцию параметры A, B, C и D;

выбрать границы отрезка [X1; X2], который содержит все возможные корни уравнения;

установить границы отрезков меньшей длины, каждый из которых содержит один корень уравнения;

определить значения корней с требуемой точностью;

**кон**

Каждая из инструкций, входящих в алгоритм (кроме задания параметров программы-функции), может быть реализована несколькими способами с помощью тех или иных программ обработки.

Среди программ обработки пакета имеется программа «КРОТ», которая вычисляет один из нулей функции. С точки зрения эффективности решения интерес представляет ее использование в сочетании с другими программами обработки. В этой связи заслуживает внимания программа «РЕКА», которая находит участки монотонности функции.

Учет особенностей вида кривой  $P(x)$  как полинома 3-й степени, наличия у нее участков монотонности, локальных экстремумов и их знаков может ускорить нахождение корней уравнения с помощью программы «КРОТ».

Рассмотрим способ решения, сочетающий возможности программ «КРОТ» и «РЕКА».

Способ А. Построим алгоритм решения задачи на применении программы КРОТ, которая находит нуль функции  $P(x)$ .

Для определения корня учащийся должен выбрать X1 и X2 так, чтобы на концах отрезка функция  $P(x)$  имела разные знаки. У данного уравнения может быть от одного до трех корней, что зависит от числа пересечений или касаний графика полинома  $P(x)$  с осью OX. Выясняя возможное положение корней, определим участки возрастания и убывания  $P(x)$  с помощью программы РЕКА.

МОТОР: РАБОТАЕМ С ФУНКЦИЕЙ

$$y(x) = Ax^3 + Bx^2 + Cx + D$$

Параметры функции: A, B, C и D

A= 4

B= -6

C= 2.25

D= -0.2

Задай параметры

67

Область определения  $P(x)$  разобьется на отрезки, которые могут содержать по одному корню. Используем границы этих отрезков как параметры настройки программы «КРОТ». Требуемой точности при необходимости достигнем повторным вызовом программы «КРОТ» на отрезке меньшей длины.

ШАГ 0. Введем в программу-функцию значения параметров  $A=4$ ,  $B=-6$ ,  $C=2,25$  и  $D=-0,2$  (рис. 8).

ШАГ 1. При анализе участков возрастания и убывания  $P(x)$  границы отрезка  $[X1; X2]$  следует выбрать так, чтобы он отражал все особенности поведения  $P(x)$ . Выберем  $X1$  и  $X2$ , исходя из их физического смысла: их значения не могут быть отрицательными величинами, а с другой стороны, они не могут превосходить длину стороны заготовки. Поэтому выберем  $X1=0$ ,  $X2=1,5$ .

ШАГ 2. Вызываем программу «РЕКА», задавая  $X1=0$ ,  $X2=1,5$ .

68

ШАГ 3. Найдены участки возрастания и убывания  $P(x)$  (рис. 5):

- на  $[0; 0,255]$   $P(x)$  возрастает;
- на  $[0,255; 0,75]$   $P(x)$  убывает;
- на  $[0,75; 1,5]$   $P(x)$  возрастает.

Точки  $X=0,255$  и  $X=0,75$  являются абсциссами локальных максимума и минимума  $P(x)$  на заданном отрезке. В зависимости от знака функции в этих точках  $P(x)$  может иметь один, два или три нуля. Однако ни один из этих отрезков не может содержать более одного нуля. Границы этих отрезков целесообразно использовать как параметры при вызове программы «КРОТ».

ШАГ 4. Вызываем программу «КРОТ», задавая  $X1=0$ ,  $X2=0,255$ .

ШАГ 5. Найден корень, равный  $0,1301$  с точностью  $\pm 0,0001$  (рис. 4). Точность удовлетворяет условию задачи. Переходим ко второму отрезку.

ШАГ 6. Вызываем программу «КРОТ», задавая  $X1=0,255$ ,  $X2=0,75$ .

ШАГ 7. Найден корень, равный  $0,3935$  с точностью  $\pm 0,0002$  (рис. 9). Точность удовлетворяет условию задачи. Переходим к третьему отрезку.

ШАГ 8. Вызываем программу «КРОТ», задавая  $X1=0,75$ ,  $X2=1,5$ .

ШАГ 9. Найден корень, равный  $0,9759$  с точностью  $\pm 0,0004$  (рис. 10). Точность удовлетворяет условию задачи. Найдены три корня уравнения:

$$XK1=0,13\pm 0,01; XK2=0,39\pm 0,01; XK3=0,98\pm 0,01$$

Завершив компьютерную часть решения задачи, вернемся к анализу полученных корней уравнения. Следует отметить, что не все три значения, хотя все они действительные, являются решением задачи. Третий корень  $XK3=0,98$  не удовлетворяет смыслу задачи, так как он больше половины ширины заготовки ( $0,98 > 0,75 = 1,5:2$ ), и такую выкройку сделать невозможно. Два других корня удовлетворяют условию и дают разные соотношения высоты и стороны основания кобова.

Отв е т. Для получения выкройки короба заданного объема требуется сделать квадратные вырезы размером  $13$  см или  $39$  см.

Работа с пакетом «МОТОР» представляет собой один из этапов решения задачи. Программы пакета при этом исполняют роль инструмента, который позволяет реализовать целый этап, т. е. решить подзадачу предложенного алгоритма. Вместе с тем применение пакета «МОТОР» не заслоняет учащемуся математического смысла выполняемых действий при решении уравнения. Начальный этап решения задачи (составление уравнения) и конечный (отбор корней) не требуют применения компьютера и выполняются так же, как и при решении любой другой задачи. Это позволяет учителю составлять план урока в соответствии с особенностями класса, текущими требованиями и дидактическими задачами, со своим личным опытом. При этом

МОТОР: работает программа КРОТ

X1= .255                      X2= .75

■ При  $X= 3.9350E-01$

Y(X) обращается практически  
в нуль!

■ Ошибка  $\Delta x$  не более  $\pm 2.E-04$

Изменить X1, X2? [Да(1), Нет(0)]

МОТОР: работает программа КРОТ

X1= .75                      X2= 1.5

■ При  $X= 9.7595E-01$

Y(X) обращается практически  
в нуль!

■ Ошибка  $\Delta x$  не более  $\pm 4.E-04$

Изменить X1, X2? [Да(1), Нет(0)]

## СПРАВОЧНИК пакета МОТОР

### РАБОТА с клавиатурой

- ? — работа со справочником для получения разъяснений.
- 1 2 3 4 5 6 7 8 9 0 + - . E — задание чисел, например, 5 -21 3.14 5E-6 -1.3E+11 7.1E10
- ← — исправление (стирание) ошибки в задаваемом числе.
- ↓ — передача заданного числа на обработку в компьютер.

Нажми левую клавишу

независимо от формы работы учащихся учитель играет активную роль в управлении ходом урока. Применение пакета МОТОР позволяет учителю организовать такой режим работы, когда учащиеся активно используют на уроке не только компьютер, но и книгу (учебник, задачник, справочник), тетрадь (конспект, рабочие записи идеи, алгоритма, промежуточных результатов и ответа к задаче) и другие средства.

Обращение к программам пакета «МОТОР» максимально упрощено. Любые действия школьника (задание параметров, выбор программы, вызов кадра справочника) осуществляются вводом числа. Поэтому учащемуся достаточно знать три формы представления чисел в ЭВМ, уметь их читать и вводить. Для этого ему требуется уметь пользоваться 16 клавишами: это цифры, точка, «E», регистр, запятой, знаки «+» и «-». При необходимости учащийся может вспомнить назначение клавиш при обращении к программам пакета, воспользовавшись одним из кадров справочника «МОТОРа» (рис. 11).

Содержательная сторона работы с пакетом требует от учащегося не составления блок-схемы алгоритма или его записи на алгоритмическом языке, а поиска способа решения задачи, умения анализировать ее математическую модель, видеть решение не как конец цепочки алгебраических преобразований, а как цель, которой можно и нужно достигнуть.

Возможности пакета позволяют реализовать целый ряд способов поиска решения поставленной задачи (см. приложение). Эти способы отличаются по трудоемкости, и их сравнение дает учащимся представление об эффективности использования ЭВМ для решения задачи.

В способах Б и В примерные границы отрезков, содержащих корни, устанавливаются с помощью программ «ОКНО» и «РАМКА». Эти же программы используются и для опре-

деления значений корней с требуемой точностью.

Программы «ОКНО» и «РАМКА» дают учащемуся полное представление о поведении функции на заданном отрезке. Между тем использование этих программ имеет свои особенности. График функции, который строит программа «ОКНО», позволяет зрительно убедиться в наличии корня. Однако, чтобы правильно выбрать границы отрезка, содержащего корень, учащийся должен прочесть график и установить, при каких значениях аргумента функция имеет тот или иной знак. С другой стороны, таблица значений, которую выдает на экран программа «РАМКА», позволяет точно установить, при каких значениях переменной функция положительна или отрицательна. Но в отличие от анализа графика, при чтении таблицы учащемуся нужно понять поведение функции по ее численным значениям.

В способе Б циклическое повторение вызова программы «РАМКА» и выбора отрезка, содержащего корень, отражает принцип последовательного уточнения корня, который используется и в программе «КРОТ». Сравнение трудоемкости способов решения задачи с использованием программы «КРОТ» и без нее раскрывает учащимся преимущества использования готовых программ.

Математический смысл способов Б и В заключается в анализе таблицы или графика функции для нахождения требуемых значений и аргументов. В этих способах отражается единство трех форм представления функциональной зависимости — аналитической, графической и табличной. В школьном курсе математики такие методы исследования функций, как анализ значений по графику и таблице, несмотря на их важность, малоэффективны из-за своей трудоемкости и значительных затрат времени на выполнение рутинных вычислительных и графических операций. Использование программ обработки «ОКНО» и «РАМКА» освобождает учителя и учащихся от непродуктивных затрат учебного времени. Работа с ними позволяет учителю дать ученикам более широкие, прочные, основанные на личном опыте знания по исследованию функций, раскрывает новые дидактические возможности изложения таких понятий вычислительной математики, как масштаб и погрешность. Опыт практических вычислений на ЭВМ формирует у учащихся правильное восприятие единства идеальной математической модели («график  $P(x)$  пересекает ось  $Ox$  в точке  $XK1$ ) и ее практического воплощения (« $P(x)$  имеет нуль при  $XK1=0,13$  с погрешностью  $\pm 0,01$ ). Оценка погрешности является неотъемлемой частью всех машинных способов решения задач.

Важным отличием пакета программ «МОТОР» от других компьютерных средств обучения является то, что он служит главным образом инструментом для решения тех или иных задач, требующих использования математического аппарата исследования функций. Пакет «МОТОР» обеспечивает учащемуся определенные удобства в работе: следит за правильностью ввода чисел, снабжает справочными материалами, предлагает возможные варианты дальнейших действий в форме меню (рис. 12), фиксирует последовательность шагов и напоминает ее по окончании работы (рис. 13), отсчитывает время работы ученика и компьютера (рис. 14). Однако «МОТОР» не предлагает учащемуся никакого сценария его общения с программами, никак не предопределяет последовательность его шагов по решению задачи. Эта особенность «МОТОРа» с одной стороны, разделяет содержание машинной и безмашинной частей компьютерного урока. С другой стороны, его подготовка и проведение имеют для учителя ряд особенностей. Одной из них является необходимость специального подбора задач, в которых, по мнению авторов, должны сочетаться три качества:

70

1. Математический аппарат, требуемый для решения задачи, должен оправдывать применение ЭВМ. Задачи, легко решаемые стандартными методами школьной математики, целесообразно использовать только для начального ознакомления с работой пакета.
2. Способ решения задачи, который выбирает учащийся при работе с пакетом «МОТОР», должен показывать ему преимущества применения ЭВМ, раскрывать особенности машинно-ориентированного подхода.
3. При составлении задач необходимо учитывать психологические особенности по-

12а

МОТОР: ПРОГРАММЫ ОБРАБОТКИ

- 1 - РАМКА дает таблицу функции
- 2 - ОКНО строит график функции
- 3 - ТРЕК находит наибольшее и наименьшее значение функции
- 4 - КРОТ находит точку пересечения функции с осью OX
- 5 - РЕКА находит участки убывания и возрастания функции
- 6 - МОРЕ вычисляет интеграл

Задай нужный номер (от 1 до 3)

МОТОР: ты можешь

- 1 - ВЫБРАТЬ ПРОГРАММУ ОБРАБОТКИ
- 2 - ИЗМЕНИТЬ ПАРАМЕТРЫ функции
- 3 - ЗАКОНЧИТЬ работу с МОТОРОМ

Задай нужный номер (от 1 до 3)

13

МОТОР напоминает о ходе работы

Шаг №	Про-грамма	X1	X2
0	A= 4.000 C= 2.250		B= -6.000 D= -0.200
2	РЕКА	0.00E+00	1.500
4	КРОТ	0.00E+00	0.255
6	КРОТ	0.255	0.750
8	КРОТ	0.750	1.500

Нажми левую клавишу

14

Мы вместе работали 4 мин 1 с  
Ты, человек, мыслил 2 м 24 с  
Я, компьютер, считал 1 м 37 с

Нажми левую клавишу

строения учащимися способов их решения на ЭВМ. От условия задачи требуется содержательность и предметная направленность, которые стимулируют интерес у учащихся как к получению ответа, так и к поиску решения.



## Выводы

1. Целью компьютерного занятия с пакетом «МОТОР» является обучение решению задач на ЭВМ на основе работы пользователя с реальным пакетом прикладных программ. Решая задачу с помощью программ пакета, учащийся обращается к реализованным в них алгоритмам не как к объектам анализа или синтеза, а как к рабочему инструменту. Работая с пакетом программ, учащийся управляет ходом решения задачи по составленному им человеко-машинному алгоритму.

2. Применение на уроке пакета «МОТОР» не требует от учащихся знаний по программированию на каком-либо языке. Это позволяет включать работу с этим пакетом в занятия на ЭВМ по разным темам курса информатики и других предметов. Правила работы учащегося с пакетом построены так, что для их освоения не требуется больших затрат учебного времени. Надежная работа программ пакета обеспечивается защитой от неверных действий учащихся.

3. Формы работы с пакетом на уроке могут быть разными, в зависимости от характера решаемой задачи: индивидуальной, бригадной или фронтальной. Сочетание различных форм работы при построении компьютерного занятия дает учителю свободу выбора необходимого метода или приема обучения в соответствии с его личным опытом и позволяет учесть особенности конкретного класса, учебной программы и других обстоятельств.

4. Работая с пакетом «МОТОР», учащиеся обращаются к реальным программам для данной конкретной ЭВМ. Это позволяет учителю расширить их представления о точности вычислений при решении задач. Оценка точности получаемых результатов развивает у учащихся грамотный подход к решению вычислительных задач на ЭВМ.

5. Важная особенность применения пакета «МОТОР» — это возможность познакомить учащихся с понятием рационального способа решения задачи, поставить вопросы о критерии оптимальности использования ЭВМ.

6. Подготовка занятий с пакетом «МОТОР» требует от учителя правильного подбора задач и грамотной организации работы учеников на уроке и вне его. Пакет «МОТОР» сопровождается набором задач и методическими рекомендациями по проведению занятий, на которых учащиеся знакомятся с работой пакета и осваивают основные особенности решения задач.

По всем вопросам программного и методического обеспечения пакета прикладных программ «МОТОР» можно обращаться в НИИ школ Минпроса РСФСР к В. В. Ани-

симвову по адресу: 109044, Москва, Крутицкий вал, 24. Тел. 276-47-32.

## Приложение

В приложении даны два способа решения рассмотренной выше задачи, раскрывающие многообразие подходов к использованию программ пакета и созданию компьютерных уроков с использованием пакета «МОТОР».

Способ Б. Используем для решения задачи программу обработки (ПО) «РАМКА», которая дает таблицу  $P(x)$ . Поскольку для любого отрезка  $[X_1; X_2]$  ПО «РАМКА» вычисляет и печатает 9 значений, план решения должен быть направлен на поиск и выделение отрезков, содержащих нули  $P(x)$ , и последующее уточнение их значений до получения требуемой точности 0,01. Признаком того, что отрезок содержит нуль функции, служат разные знаки  $P(x)$  на его концах.

ШАГ 0. Введем в программу-функцию значения параметров  $A=4$ ,  $B=-6$ ,  $C=2,25$ ,  $D=-0,2$ .

ШАГ 1. Выберем отрезок для анализа значений  $P(x)$  так, чтобы ни одно из решений задачи не оказалось вне его границ. Допустим,  $X_1=0$ ,  $X_2=1,5$ . Поскольку данные границы выбраны исходя из физического смысла переменной  $x$  (см. способ А), рассмотренные  $P(x)$  вне их нецелесообразно.

ШАГ 2. Вызываем программу «РАМКА» при  $X_1=0$ ,  $X_2=1,5$ .

ШАГ 3. Получаем таблицу (рис. 1), анализ которой позволяет установить три отрезка меньшей длины, на которых функция меняет знак:

на отрезке  $[0; 0,188]$   $P(x)$  меняет знак с отрицательного

$(P(0)=-0,2)$  на положительный

$(P(0,188)=0,037)$ ;

на отрезке  $[0,375; 0,563]$   $P(x)$  меняет знак с положительного

$(P(0,375)=0,01)$  на отрицательный

$(P(0,563)=-0,12)$ ;

на отрезке  $[0,938; 1,13]$   $P(x)$  меняет знак с отрицательного

$(P(0,938)=-0,07)$  на положительный

$(P(1,13)=0,43)$ .

Уточним значение корня, принадлежащего отрезку  $[0; 0,188]$ . Выберем  $X_1=0$ ,

$X_2=0,188$ .

ШАГ 4. Вызываем программу «РАМКА» для  $X_1=0$  и  $X_2=0,188$ .

ШАГ 5. По таблице (рис. 15) устанавливаем, что  $P(x)$  меняет знак на отрезке  $[0,12; 0,14]$  ( $P(0,12)=-0,01$ ,  $P(0,14)=0,009$ ). Длина отрезка равна 0,02, т. е. удвоенной допустимой погрешности. Поэтому в качестве значения первого корня с требуемой точностью принимаем среднюю величину

МОТОР: работает программа РАМКА  
 $X1 = 0$        $X2 = .188$

X	Y(X)
$X1 = 0$	-0.2000
.0235	-0.1504
.047	-0.1071
.0705	-0.0698
.094	-0.0382
.1175	-0.0120
.141	9.177E-03
.1645	0.0256
$X2 = .188$	0.0375

Изменить X1, X2? [Да(1), Нет(0)]

0,13. Переходим к уточнению другого корня на отрезке [0,375; 0,563].

72

Процесс уточнения корня представляет повторение трех действий: а) выполнение программы «РАМКА» для получения таблицы значений  $P(x)$  на заданном отрезке; б) нахождение границ отрезка меньшей длины, содержащего корень; в) сравнение длины отрезка с требуемой погрешностью и принятие решения о прекращении или продолжении уточнения корня. В выполняемых действиях выявляется цикличность, поэтому для записи действий по уточнению другого корня воспользуемся сокращенным описанием шагов.

ШАГ 6. Вызов ПО «РАМКА» на [0,375; 0,563].

ШАГ 7. Выбор отрезка [0,380; 0,403] ( $P(0,380) = 0,008$ ,  $P(0,403) = -0,006$ ). Длина отрезка  $> 0,02$  — продолжаем уточнение.

ШАГ 8. Вызов ПО «РАМКА» на [0,380; 0,403].

ШАГ 9. Выбор отрезка [0,392; 0,394] ( $P(0,392) = 0,001$ ,  $P(0,394) = -0,0005$ ). Длина отрезка  $< 0,02$  — прекращаем уточнение. С требуемой точностью принимаем значение второго корня равным 0,39. Переходим к уточнению значения последнего корня, который принадлежит отрезку [0,94; 1,13].

ШАГ 10. Вызов ПО «РАМКА» на [0,94; 1,13].

ШАГ 11. Выбор отрезка [0,975; 0,979] ( $P(0,975) = -0,003$ ,  $P(0,979) = 0,005$ ). Длина отрезка  $< 0,02$  — принимаем третий корень равным 0,98.

Найдены все три корня уравнения:  $XK1 = 0,13 \pm 0,01$ ;  $XK2 = 0,39 \pm 0,01$  и  $XK3 = 0,98 \pm 0,01$ .

Способ В. Программа «ОКНО» реализует в пакете графический способ представления функции. Анализ графика позволяет установить интересующие значения функции и соответствующие значения аргумента. Выбирая границы «окна», в котором строится гра-

фик, учащийся может определить значения нулей функции с любой точностью. При работе с программой «ОКНО» необходимы анализ точности получаемых по графику значений и проверка их соответствия условиям задачи.

ШАГ 0. Введем в программу-функцию значения параметров  $A=4$ ,  $B=-6$ ,  $C=2,25$ ,  $D=-0,2$ .

ШАГ 1. Зададим границы «окна» по оси  $Ox$  так, чтобы они заведомо содержали все решения задачи. Поскольку переменная  $x$  обозначает длину, необходимое условие  $x > 0$ . С другой стороны, значение  $x$  не может превосходить размер заготовки, т. е.  $x < 1,5$ . Выберем  $X1=0$ ,  $X2=1,5$ , так как рассмотрение  $P(x)$  вне отрезка [0; 1,5] не имеет смысла.

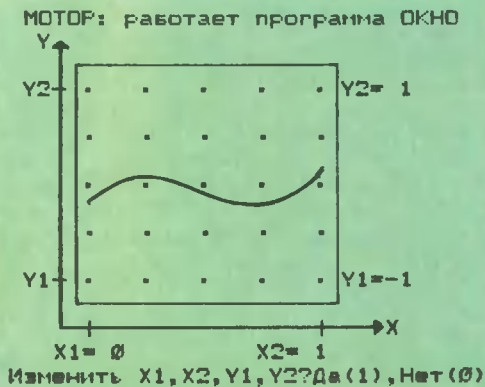
Установим границы по оси  $Oy$  так, чтобы они позволили получить общее представление о поведении  $P(x)$  на всем отрезке [ $X1$ ;  $X2$ ]. Для удобства определения нулей выберем  $Y1$  и  $Y2$  симметрично от начала координат. Возьмем  $Y1=-10$ ,  $Y2=10$ .

ШАГ 2. Вызываем программу «ОКНО» при  $X1=0$ ,  $X2=1,5$ ,  $Y1=-10$ ,  $Y2=10$ .

ШАГ 3. Получаем график (рис. 16), который дает представление об общем виде кривой  $P(x)$  — кубической параболы. На нем различаются локальные экстремумы и один из нулей. Выбранный масштаб не позволяет установить наличие двух других нулей или их местоположение. Однако из графика видно, что вне отрезка [0; 1] функция не меняет знак и не может иметь нулей. Рассмотрим поведение  $P(x)$  на указанном отрезке. Поскольку требуется найти точки пересечения графика функции с осью  $Ox$ , от выбора границ по оси  $Oy$  зависит наглядность выявления этих точек на графике. Поскольку важно установить число корней, выберем  $Y1=-1$ ,  $Y2=1$ .

ШАГ 4. Вызываем программу «ОКНО» для  $X1=0$ ,  $X2=1$ ,  $Y1=-1$ ,  $Y2=1$ .





ШАГ 5. Получаем график (рис. 17), на котором видны все три корня уравнения. Масштаб позволяет установить, что корни принадлежат следующим трем отрезкам:  $[0; 0,2]$ ;  $[0,2; 0,6]$ ;  $[0,8; 1]$

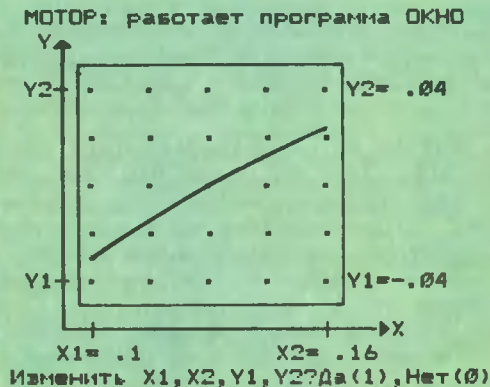
Таким образом, удалось локализовать все три корня.

Определим значения корней с требуемой точностью. Найдем нуль  $P(x)$  на отрезке  $[0; 0,2]$ . Значения  $Y1$  и  $Y2$  будем выбирать так, чтобы график  $P(x)$  пересекал ось  $OX$  под углом, близким к прямому. Это сделает чтение графика более удобным и ускорит решение задачи. Возьмем  $Y1 = -0,1$ ,  $Y2 = 0,1$ .

ШАГ 6. Вызываем программу «ОКНО» с  $X1 = 0$ ,  $X2 = 0,2$ ,  $Y1 = -0,1$ ,  $Y2 = 0,1$ .

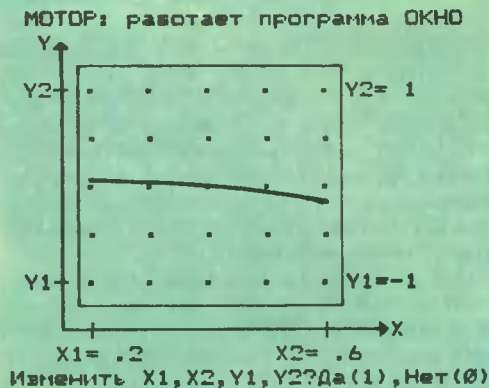
ШАГ 7. По полученному графику (рис. 18) уточняем местоположение корня. Он принадлежит отрезку  $[0,10; 0,16]$ , который намного меньше отрезка  $[0; 0,2]$ . Так как требуемая точность еще не достигнута, получим график для найденного отрезка, уменьшив границы и по оси  $OY$ .

ШАГ 8. Вызываем программу «ОКНО» с  $X1 = 0,1$ ,  $X2 = 0,16$ ,  $Y1 = -0,04$ ,  $Y2 = 0,04$ .

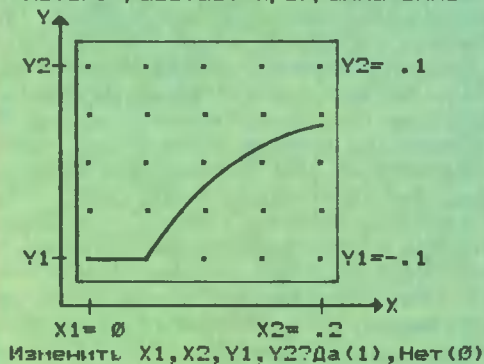


ШАГ 9. Получаем график (рис. 19), по которому значение искомого корня может быть установлено с требуемой точностью. Оно равно  $0,13$ . Переходим к поиску второго корня на отрезке  $[0,2; 0,6]$ .

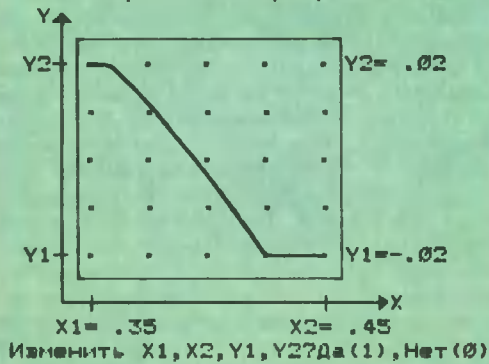
ШАГ 10. Вызываем программу «ОКНО» для  $X1 = 0,2$ ,  $X2 = 0,6$ ,  $Y1 = -1$ ,  $Y2 = 1$ .



MOTOR: работает программа ОКНО



MOTOR: работает программа ОКНО



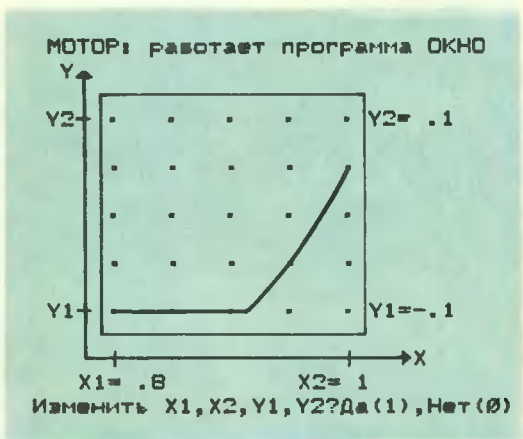
ШАГ 11. Получаем график (рис. 20), который позволяет указать местоположение искомого корня более того. Корень принадлежит отрезку  $[0,35; 0,45]$ . Посмотрим график на найденном отрезке. Границы по оси  $OY$  сократим, как и при поиске первого корня.

ШАГ 12. Вызываем программу «ОКНО» с  $X1=0,35$ ,  $X2=0,45$ ,  $Y1=-0,02$ ,  $Y2=0,02$ .

ШАГ 13. По полученному графику (рис. 21) находим значение второго корня. С требуемой точностью оно равно  $0,39$ . Переходим к определению третьего корня на отрезке  $[0,8; 1]$ .

ШАГ 14. Вызываем программу «ОКНО» с параметрами настройки  $X1=0,8$ ,  $X2=1$ ,  $Y1=-0,1$ ,  $Y2=0,1$ .

ШАГ 15. Получаем график (рис. 22), масштаб которого позволяет с достаточной точностью определить значение последнего корня, равное  $0,98$ .



Задача решена. Найдены три корня кубического уравнения:  $XK1=0,13\pm 0,01$ ;  $XK2=0,13\pm 0,01$ ;  $XK3=0,98\pm 0,01$ .

А. САВЧЕНКО, А. ФИЛИППОВ, К. КОЛОСОВ, В. ПОЛЕТАЕВ, П. ФЕДОРОВ,  
А. ПАСХИН

## Разработка обучающих систем

Возрастание объема информации на современном этапе научно-технической революции требует введения непрерывного образования и дальнейшей его интенсификации. Использование ЭВМ помогает сделать индивидуальным процесс обучения.

При автоматизированном обучении и контроле знаний распространены два способа применения ЭВМ: использование языков программирования (Бейсик, Фортран, Паскаль и др.) для создания обучающих и контролирующих программ; использование автоматизированных обучающих систем (АОС) для разработки автоматизированных обучающих курсов (АОК).

Языки программирования из-за своей универсальности не обеспечивают уровня сервиса для эффективной разработки и использования АОК. В то же время АОС предоставляет средства, позволяющие автору АОК реализовать адекватную модель своей предметной области с использованием базы знаний АОС. Средства сбора и обработки статистики процесса обучения в функционально развитых системах могут использоваться для управления им. В отличие от языков программирования, АОС можно рассматривать как средство автоматизации проектирования АОК, позволяющее решать задачи обучения на качественно новом уровне. При разработке неболь-

шого числа обучающих курсов экономически оправдан первый способ применения ЭВМ, а при комплексном использовании АОК наиболее эффективными являются автоматизированные обучающие системы.

Особую значимость АОС получают при решении задачи обеспечения всеобщей компьютерной грамотности, которая зачастую понимается как повсеместное обучение языкам программирования. По мнению авторов, программирование является такой же сферой профессиональной деятельности, как, например, конструирование машин. Поэтому компьютерную грамотность надо рассматривать как умение работать с ЭВМ в сфере профессиональных интересов пользователя. Для учебных заведений — это использование компьютера при изучении курсов математики, биологии, иностранного языка. Автоматизированные обучающие системы обладают необходимыми возможностями для построения эффективных АОК по различным дисциплинам. При этом один из классов АОС — так называемые авторские системы — позволяют создавать АОК специалистам гуманитарного профиля с такой же легкостью, как и инженерам. Таким образом, АОС является более эффективным (по сравнению с языками программирования) средством достижения компьютерной грамотности.

Используемая техническая база влияет на распространенность и эффективность применения АОС. Так, основной предпосылкой распространенности АОС является использование стандартного оборудования и его стоимость. При этом начальные затраты на приобретение большой ЭВМ превышают затраты на класс персональных ЭВМ. Замена класса ПЭВМ на локальную сеть микро-ЭВМ, использующую терминальные машины без периферийного оборудования, приводит к дальнейшему снижению стоимости технического обеспечения, но одновременно и к уменьшению функциональных возможностей. При этом соотношение стоимость — педагогическая эффективность таких классов оказывается лучше, чем у классов, построенных на базе персональных микро-ЭВМ. Это связано с возможностью использования микро-ЭВМ с большим объемом ОЗУ, накопителем на жестком диске и др. в качестве компьютера преподавателя, а рабочие места обучаемых можно оснастить менее дорогими микро-ЭВМ, но с расширенным комплектом внешних устройств, среди которых графические приставки, цветные мониторы, музыкальные синтезаторы, световое перо. Сетевое программное обеспечение должно предусматривать доступ периферийных микро-ЭВМ к внешним устройствам головной машины.

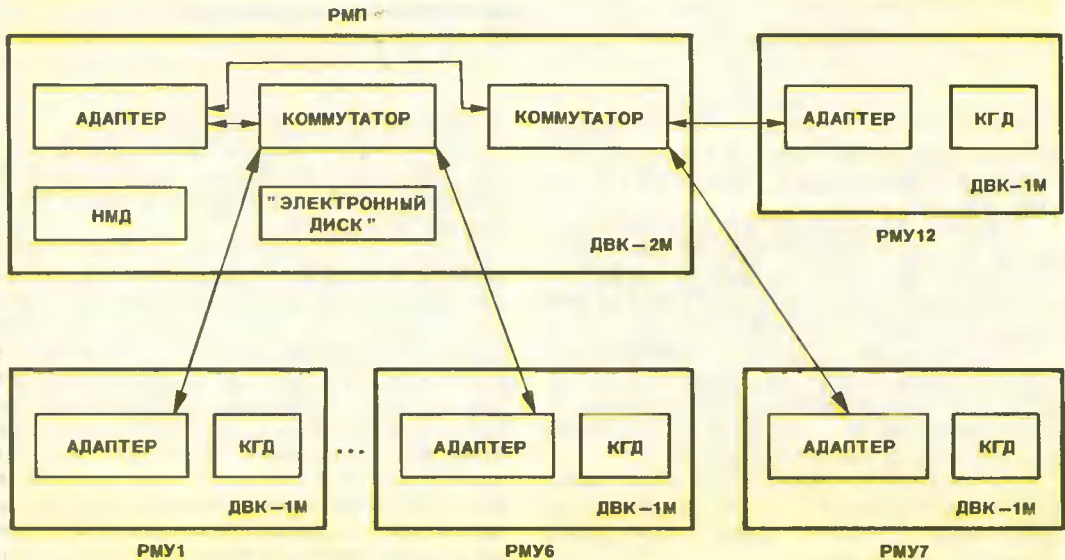
Создание АОС с широкими возможностями для микро-ЭВМ, имеющей ограниченные ресурсы, предполагает разделение программы на основной и оверлейный сегменты. Для функционирования такой АОС необходима загрузка операционной системы в каждую периферийную микро-ЭВМ, что в свою оче-

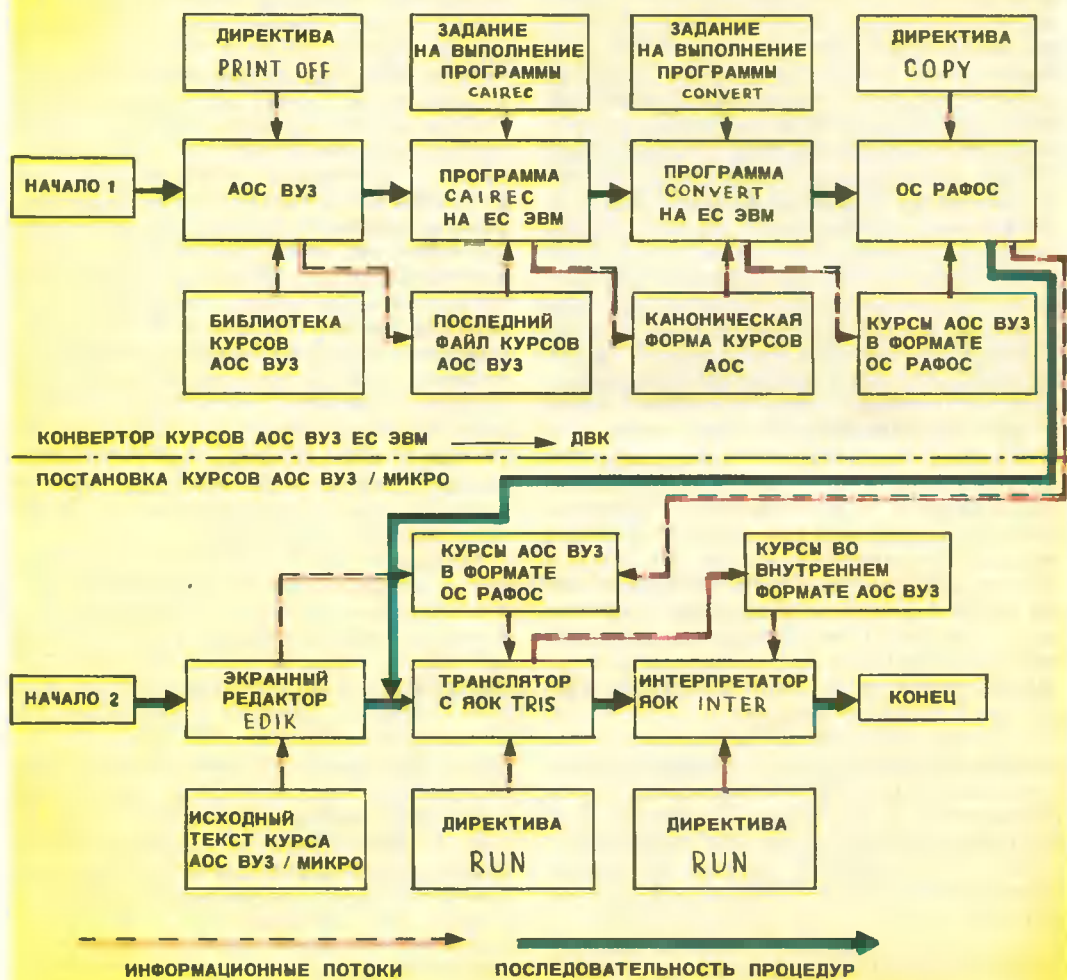
редь требует разработки сетевых адаптеров с высокой скоростью передачи данных.

Сетевое программное обеспечение должно обеспечивать такой же режим работы пользователей с коллективными внешними устройствами, как и с обычными, подключенными непосредственно к микро-ЭВМ. В этом случае программное обеспечение для АОС, разработанное и отлаженное на персональной ЭВМ, без изменений может быть использовано для работы в составе сети. По такому принципу действует малая локальная сеть, созданная в вычислительном центре Московского института электронной техники (рис. 1). Скорость передачи данных в сети — 0,5 Мбит/с. Сеть испытана для двух обучающих систем, созданных в институте: автоматизированной обучающей системы для вузов АОС ВУЗ/МИКРО-1 и автоматизированной системы диалогового обучения АСТРА/МИКРО. Данный аппаратно-программный комплекс получил название вычислительного обучающего модуля (ВОМ-1).

Система АОС ВУЗ/МИКРО-1 предназначена для разработки курсового обеспечения и использования курсов в режиме обучения — контроля с использованием в качестве рабочего места учащегося (РМУ) ДВК-2М и ДВК-3. В качестве рабочего места преподавателя (РМП) может быть использована ДВК «Электроника-100/25», СМ-1420. Система совместима на уровне языка обучающих курсов (ЯОК) с базовой АОС ВУЗ для ЕС ЭВМ.

Основная задача при разработке системы АОС ВУЗ/МИКРО-1 — создание средства, обеспечивающего интерпретацию курса, разработанного на ЕС ЭВМ и перенесенного с по-





мощью конвертора в среду микро-ЭВМ. Созданная технология переноса курсового обеспечения предполагает следующие этапы (рис. 2):

1. Вывод содержимого курса в текстовой форме из среды базовой АОС ВУЗ/ЕС в последовательный набор данных на диске.
2. Конвертирование исходного текста курса на магнитную ленту в формате СМ ЭВМ.
3. Копирование курса средствами операционной системы РАФОС на гибкий диск на СМ ЭВМ или мини-ЭВМ типа «Электроника» с подключенными накопителями на магнитной ленте и гибком магнитном диске.
4. Обработка на ДВК исходного текстового файла транслятором с ЯОК. В результате трансляции создается файл с внутренней формой представления.
5. Интерпретация курса с помощью интер-

претатора ЯОК в режиме обучения.

При условии непосредственного подключения ДВК к ЕС ЭВМ процесс переноса курсов упрощается за счет исключения этапов 2 и 3 данной технологии.

Кроме использования курсов с ЕС ЭВМ, имеется возможность разрабатывать курсовое обеспечение непосредственно на ДВК.

Рассмотрим процесс разработки курса с использованием системы АОС ВУЗ/МИКРО. После создания сценария курса, формализации учебного материала, разбиения его на вопросы автор курса пишет программу на ЯОК, которая вводится с помощью символического редактора в файл на гибком магнитном диске. Вызывается транслятор с ЯОК, который обрабатывает исходную программу. При этом производится синтаксический анализ и выдаются диагностические сообщения об

ошибках. Если исходная программа их не содержит, то создается файл с внутренней формой представления. Интерпретатор обеспечивает последовательное считывание и выполнение операторов ЯОК. В его состав входит функциональный отладчик, обеспечивающий трассировку выполнения операторов и вывод содержимого специальных областей памяти.

Процесс разработки курсового обеспечения с использованием АОС ВУЗ/МИКРО во многом аналогичен традиционному циклу отладки программ. Функциональные возможности ЯОК позволяют создавать курсы с адаптивными свойствами. В состав системы дополнительно входят средства подключения внешних функций, написанных на языке Паскаль. Так пользователь может расширять систему за счет включения специализированных функций, например вычислений, управления лабораторным оборудованием и т. п.

Разработка курсов на ЯОК трудоемка, требует квалификации пользователя в области программирования на этапе составления курса с использованием ЯОК и в процессе его отладки. Это затрудняет освоение начинающим разработчиком АОК принципов автоматизированного обучения. Одно из направлений решения этой проблемы — введение в систему средств автоматизации проектирования и разработки курсового обеспечения. Их введение планируется в последующих версиях системы АОС ВУЗ/МИКРО.

Автоматизированная система диалогового обучения (АСДО) АСТРА/МИКРО относится к классу авторских систем, ориентированных на пользователя — непрофессионала в области вычислительной техники и программирования, что особенно актуально для профессиональной и общеобразовательной школы. Система является «дружественной» к пользователю и содержит средства обеспечения диалога, основанные на принципах «меню» и «интервью» с автором курса. Прототипом данной системы является АСДО АСТРА для ЕС ЭВМ.

Процесс разработки курса представляет собой диалог автора курса с системой. При этом она задает вопросы в такой последовательности, которая позволяет начинающему пользователю, еще плохо знакомому с клавиатурой и принципами работы ЭВМ, почувствовать уверенность в правильности выполняемых действий. Время освоения системы пользователем не превышает трех часов, и, как правило, за это время он успевает подготовить небольшой фрагмент курса.

В системе АСТРА/МИКРО не используется авторский язык, аналогичный ЯОК АОС ВУЗ. Весь процесс разработки и проведения курса осуществляется в диалоговом режиме без использования сервисных средств опера-

ционной системы. С точки зрения пользователя ДВК, АСТРА/МИКРО — индивидуальное средство разработки курсового обеспечения, обучения и самообучения. При этом не требуется знать команды операционной системы, так как загрузка обучающей системы осуществляется автоматически. В ее состав входят сервисные средства распечатки содержимого курса, копирования и восстановления курсов.

Для автоматизации проектирования курсового обеспечения и упрощения формализации учебного материала автору предлагаются АСТРА-бланки, с помощью которых можно представить структуру курса в виде граф-схемы. На бланках записываются и вопросы. При этом сообщения системы в процессе ввода курса содержат названия полей бланка.

В состав системы входят средства обеспечения машинной графики для ДВК-3. Это графический редактор и программа распечатки графического изображения. С помощью графического редактора можно формировать произвольные изображения и шрифты. Редактор включает знакогенератор, обеспечивающий формирование букв различной высоты и под различными углами, и подсистему автоматизированного построения элементарных фигур. Средства машинной графики дают возможность дополнительно использовать систему для обучения техническим дисциплинам и иностранным языкам.

Система АСТРА/МИКРО передана в 80 организаций страны и отраслевой фонд алгоритмов и программ НПО «Центрпрограммистем» (г. Калинин) и используется для обучения информатике, электротехнике, технической механике, иностранному языку, технике безопасности, медицине и другим дисциплинам. Она принята в качестве базовой в системе профессионально-технического образования страны. Например, система активно используется во ВНИИ профессионально-технического образования и СПТУ № 38 Ленинграда.

Системы АОС ВУЗ/МИКРО и АСТРА/МИКРО поставляются пользователям вместе с документацией в виде копий на гибких магнитных дисках.

В процессе реализации обучающих систем были решены проблемы системного математического обеспечения, представляющие самостоятельный интерес. С использованием языка Паскаль на базе ЕС ЭВМ разработана макротехнология создания мобильного программного обеспечения, в частности получены положительные результаты при переносе систем на ЕС ЭВМ и микро-ЭВМ ЕС-1840. Разработаны драйверы, восьмибитный и много-терминальный варианты компилятора с языка Паскаль.

Для создания банка курсов и обеспечения

авторского надзора в состав программного обеспечения включены средства защиты от несанкционированного копирования. Маркерный диск, обеспечивающий получение только заданного числа копий, поставляется при условии активного применения системы пользователем.

При общей ориентации обучающих систем АОС ВУЗ/МИКРО и АСТРА/МИКРО на применение в вузах, институтах повышения квалификации и отделах переподготовки специалистов промышленности их использование в составе сети открывает дополнительные возможности для применения в общеобразовательных школах и ПТУ. Это возможно благодаря снижению стоимости средств вычислительной техники за счет применения терминальных микро-ЭВМ без внешней памяти. Такого рода классы ПЭВМ даже при относительно низкой

скорости передачи данных (9,6К бит/с) хорошо себя зарекомендовали при обучении основам информатики учащихся общеобразовательных школ с помощью программ-тренажеров.

Описанные выше системы можно использовать и как инструментальные для разработки курсового обеспечения ЭВМ типа БК-0010. Для обеспечения такой возможности был реализован вариант режима обучения на БК-0010 и ДВК-1 на языке ассемблера. При этом микрокомпьютер используется в качестве «проигрывателя» курсов, а программное обеспечение и АОК должно целиком располагаться в его ОЗУ (16 и 48К байт) соответственно.

С учетом имеющегося опыта применения функциональные возможности этих систем получают дальнейшее развитие в новых их версиях.

В. КОРНЮШКО, В. АВДЕЕВ, Г. ФРОЛОВ, А. ЖЕДЬ

## Распределенная вычислительная система МИТХТ

Выбор оптимальной системы вычислительных средств на основе персональных ЭВМ для оборудования учебного класса сопряжен с необходимостью учета большого количества разнообразных факторов, таких, например, как предмет обучения; степень сложности и необходимый уровень освоения; контингент учащихся (начальное, среднее, высшее образование, специализация и профессиональная направленность); количество, качество, функциональные возможности средств вычислительной техники (аппаратуры и программного обеспечения) и их стоимость; серийность производства аппаратуры промышленностью и др.

На кафедре вычислительной математики и ЭВМ Московского института тонкой химической технологии разработана и используется в учебном процессе распределенная вычислительная система трех конфигураций (РВС-1, 2, 3), структурные схемы которых приведены на рисунке.

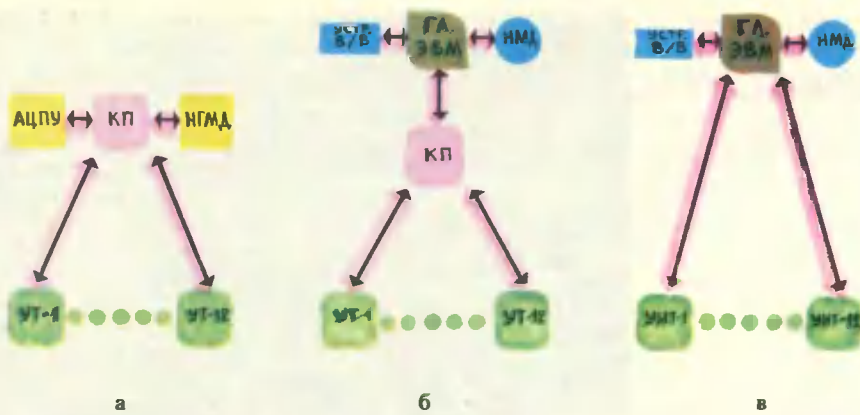
РВС МИТХТ реализована на базе серийно выпускаемых отечественной промышленностью микро-ЭВМ ДВК-1, 2, 3 и мини-ЭВМ СМ-3. Разнообразие конфигураций обусловлено особенностями организации учебного процесса для различных групп пользователей: степенью сложности предмета обучения и необходимым уровнем его освоения.

В качестве рабочего места учащегося в РВС-1, 2 используется учебный терминал (УТ) ДВК-1, 2, 3, снабженный цветным графическим монитором, который выполнен на базе бытового цветного телевизора, а РВС-3 использует учебно-исследовательский терминал (УИТ), который помимо оборудования УТ имеет накопитель на гибких магнитных дисках (НГМД) и алфавитно-цифровое печатающее устройство (АЦПУ).

РВС-1 (рис. 1, а) наиболее простая конфигурация, предназначенная для обучения студентов начальных курсов основам программирования (на Бейсике и Фокале) и для обеспечения самостоятельной работы. Существенное повышение интереса к предмету и облегчение его усвоения достигается широким использованием цветной графики. Для этого в состав интерпретатора языка Бейсик введены специальные программы, позволяющие достаточно легко выполнять разнообразные цветные графические построения. РВС-1 может быть рекомендована для оборудования учебных классов школ и других учебных заведений.

РВС-2 (рис. 1, б) предназначена для обучения студентов средних курсов языкам программирования высокого уровня (Фортран, Паскаль, ПЛ/1), для изучения методов вычислительной математики, где программы





1 а б в

каждого студента хранятся во внешней памяти в достаточно большом объеме, для автоматизированного обучения и контроля, для организации самостоятельной работы. Конфигурация РВС-2 отличается тем, что с помощью байтового параллельного интерфейса коммуникационный процессор (КП) подключается к мини-ЭВМ СМ-3.

РВС-3 (рис. 1, в) предназначена для работы студентов старших курсов при курсовом и дипломном проектировании, для проведения научно- и учебно-исследовательских работ, освоения активных методов обучения. Каждый УИТ имеет память на НГМД для хранения информации по конкретному проекту и прямой доступ к памяти на жестких дисках, где хранится справочная информация для проектирования по всем темам.

Общей конструктивной особенностью всех конфигураций является звездообразное соединение локальной сетью двенадцати терминалов с помощью двух плат контроллеров телеграфного канала (КТЛК) и интерфейсов последовательного ввода-вывода (ИПВВ); ИПВВ расположены в терминалах, а КТЛК — в зависимости от конфигурации: для РВС-1, 2 — в коммуникационном процессоре (КП), а для РВС-3 — в главной ЭВМ.

Для звездообразной структуры локальной сети конфигураций РВС-1,3 характерна последовательная асинхронная передача информации со скоростью 57,6 к бод по витым парам проводов, а для иерархической структуры типа «дерево» конфигурации РВС-2 между главной ЭВМ и КП осуществляется байтовая параллельная передача информации со скоростью 200К байт/с по плоскому кабелю. Обнаружение и исправление ошибок передачи информации по локальной сети осуществляется организацией посылок в блоки с заголовком и контрольной суммой с повтором передачи при обнаружении ошибки контрольной суммы.

Основная задача РВС МИТХТ — обес-

печение учебного процесса каждой группы пользователей оптимальным количеством оборудования и программных средств. Это достигается путем реализации принципа коллективного пользования дорогостоящим внешним оборудованием общего назначения на основе равноправного доступа к системному математическому обеспечению и индивидуального к рабочим областям пользователей.

Программное обеспечение включает базовое (системное), сетевое и прикладное (помимо стандартного). Базовая операционная система (ОС) RT 11V5.0 является наиболее доброжелательной и доступной для неквалифицированного пользователя. Сетевое обеспечение РВС-1 включает программу ICS и драйвер TS, а РВС-2,3 — программы ICH, ICM и драйверы SX, SV. Прикладное обеспечение:

- сервисная система «меню»;
- пакеты программ цветной графики для:
  - а) интерпретатора языка Бейсик,
  - б) резидентной микросистемы обучения языку ассемблера;
- пакеты программ для обеспечения лабораторных работ;
- обучающие, контролирующие программы по различным курсам и дисциплинам.

Основные достоинства программного обеспечения системы РВС МИТХТ:

1. Каждый пользователь получает в свое распоряжение полноценную систему RT11V5.0, способную выполнять все системные команды.
2. Загрузка ОС в компьютер нижнего уровня осуществляется с использованием аппаратного начального загрузчика, имеющегося в ЭВМ ДВК, и не требует дополнительных аппаратных средств типа ПЗУ.

Системой РВС МИТХТ оборудованы два дисплейных класса (см. фото) на базе ДВК-1 с цветной графикой, реализованной на бытовом телевизоре «Электроника-430», и на базе ДВК-3 с черно-белой графикой.

Дисплейный класс  
с цветной графикой,  
оборудованный  
системой РВС МИТХТ



Уже существующие дисплейные классы на базе микро-ЭВМ ДВК также можно оборудовать системой РВС МИТХТ.

В настоящее время ведется работа по созданию перспективного, с нашей точки зрения на ближайшее десятилетие, дисплейного класса для вузов в следующей комплектации:

микро-ЭВМ «Электроника УКНЦ» —

12 шт. (учебные терминалы);

микро-ЭВМ «Электроника-85» с накопителем типа «Винчестер» — 1 шт. (главная ЭВМ);

адаптер последовательной связи на 12 каналов — 1 шт.;

операционная система РВС МИТХТ.

Тел. 246-46-50. Адрес: 119831, Москва, Г-435, М. Пироговская, 1, МИТХТ.

## А. ГИГЛАВЫЙ

# Новый персональный компьютер «Архимед»

В конце 1985 г. в печати появились сообщения о совместном проекте фирм «Оливетти» (Италия), «Эйкорт» (Великобритания) и «Филипс» (Нидерланды) по созданию массового персонального компьютера нового поколения. Результатом проекта стал описанный в августовском номере журнала «Personal Computer World» за 1987 г. персональный компьютер «Архимед». Его отличительные черты — использование 32-разрядного микропроцессора ARM, созданного по проекту фирмы «Эйкорт», разнообразие конфигураций (5 моделей) и низкая стоимость (в минимальной конфигурации, без монитора — 799 фунтов стерлингов). ARM не самый быстродействующий в своем классе, но этот недостаток вполне искупают его малая стоимость и простота сопряжения с дешевыми интегральными схемами памяти.

Даже в минимальной конфигурации емкость ОЗУ ПЭВМ «Архимед» — 512К байт, что позво-

ляет работать с графическими объектами больших размеров (максимальная память кадра цветного изображения —  $640 \times 512 \times 4$  бит, в монохромном режиме — до  $1024 \times 1024$  бит) и пользоваться наиболее развитыми системами программирования — Лисп, Пролог, Си. Созданная специально для этой ПЭВМ БИС контроллера видеорежима обеспечивает работу с палитрой из 4096 цветовых оттенков и различными типами мониторов, включая мониторы высшего класса. Синтезатор звуковых эффектов использует 8-канальный синтез с обеспечением стереоэффекта. В целом возможности графики, цвета и звука «Архимеда», безусловно, находятся на уровне лучших мировых достижений в классе бытовых и учебных персональных компьютеров конца 80-х гг.

Большое внимание уделено эргономичности машины. Клавиатура повторяет набор основных и функциональных клавиш, принятый для ПЭВМ

класса IBM PC (привычности отдано предпочтение перед нововведениями). В базовую конфигурацию входит манипулятор типа «мышь». Применен малозумный вентилятор.

Итак, разработчиками «Архимеда» предпринята попытка создать массовую ПЭВМ, перекрывающую по своим показателям весь спектр традиционных применений. Успех такой попытки всецело зависит

от уровня программного обеспечения, и здесь новой машине придется выдержать серьезную конкуренцию с такими ПЭВМ, как «Амига» и «Атари 1040 ST». Возможно, что на первых порах спрос на ПЭВМ «Архимед» будет определяться наличием режима совместимости с учебными ПЭВМ «Эй-корн», для которых накоплен значительный фонд педагогических программных средств.

## Обслуживание и ремонт вычислительной техники

Типовой договор на ввод в эксплуатацию, техническое обслуживание и ремонт вычислительной техники в средних учебных заведениях в гарантийный и послегарантийный периоды ее эксплуатации утвержден в октябре 1987 г.

Договор регулирует отношения между заказчиками — средними учебными заведениями (РОНО, представляющие школы, СПТУ, техникумы) и исполнителями — предприятиями (объединениями) ГКВТИ СССР. На основании Типового договора заказчик и исполнитель заключают конкретный договор, проект которого высылается исполнителем заказчику в 20-дневный срок после получения заявки на проведение работ.

В роли заказчика для общеобразовательных школ выступает районный отдел народного образования (РОНО). Конкретный договор между РОНО и предприятием ГКВТИ заключается с учетом всех школ района, имеющих или планирующих установку КУВТ, или отдельно по каждой школе. Выполнение некоторых обязательств, предусмотренных пунктами договора, РОНО может возлагать на школы. В частности, школы должны своевременно подготовить помещение под кабинет вычислительной техники (КВТ), при этом следует руководствоваться методическими рекомендациями «Кабинет вычислительной техники всех типов средних учебных заведений» (НИИ ШОТСО АПН СССР, 1986).

На условиях, согласованных с заказчиком, исполнитель может выполнить и электромонтажные работы в помещении, отведенном под КВТ (осуществить проводку силовых кабелей до щита, укладку провода в трубы, выполнить работы по заземлению и др.).

### ТИПОВОЙ ДОГОВОР

на ввод в эксплуатацию, техническое обслуживание и ремонт вычислительной техники в средних учебных заведениях в гарантийный и послегарантийный периоды ее эксплуатации<sup>1</sup>

(наименование предприятия (объединения),

предоставляющего услуги по вводу в эксплуатацию,

<sup>1</sup> Утвержден Государственным арбитражем СССР.

техническому обслуживанию и ремонту

вычислительной техники)

в лице \_\_\_\_\_

(должность, фамилия, имя, отчество)

действующего на основании \_\_\_\_\_

(устава,

положения и т. д.)

именуемое в дальнейшем «Исполнитель», с одной стороны, и \_\_\_\_\_

(наименование районного (городского)

отдела (управления) народного образования, среднего

профессионально-технического училища, среднего спе-

циального учебного заведения, пользующегося

услугами)

в лице \_\_\_\_\_

(должность, фамилия, имя, отчество)

действующего на основании \_\_\_\_\_

(устава,

положения и т. д.)

именуемое в дальнейшем «Заказчик», с другой стороны, заключили настоящий договор о нижеследующем:

### I. Общие положения

1. «Исполнитель» обязуется осуществить ввод в эксплуатацию, техническое обслуживание и ремонт<sup>2</sup> вычислительной техники<sup>3</sup> в гарантийный и послегарантийный периоды ее эксплуатации. «Заказчик» обязуется принять и оплатить надлежаще выполненные работы.

2. Наименования техники и выполняемых работ определяются в перечне техники согласно приложению № 1 к договору.

Сроки выполнения работ, предусмотренные в перечне техники, устанавливаются в графике выполнения работ<sup>4</sup> согласно приложению № 2 к договору.

Приложения № 1 и 2 являются неотъемлемой частью договора.

<sup>2</sup> В дальнейшем — «выполнение работ».

<sup>3</sup> В дальнейшем — «техника».

<sup>4</sup> В дальнейшем — «график».

При согласовании перечня техники и графика стороны руководствуются требованиями эксплуатационной и ремонтной документации предприятия — изготовителя техники<sup>5</sup>.

3. Работы, не включенные в график, но связанные с устранением неисправностей техники, вызванных отказами или сбоями, проводятся «Исполнителем» на основании заявки (письма, телеграммы, телетайпограммы, телефонограммы) «Заказчика». В заявке указываются тип, заводской номер и дата изготовления техники, наименование «Изготовителя», период времени, в течение которого эксплуатируется техника.

4. На условиях, согласованных с «Заказчиком», «Исполнитель» выполняет для него электромонтажные работы в кабинетах вычислительной техники за дополнительную плату.

5. «Заказчик» ведет журнал по учету рабочего состояния техники и представляет его «Исполнителю» для внесения записей о выполненных работах. Записи в журнале учитываются при составлении актов о невыполнении или некачественном выполнении работ.

6. При заключении договора стороны могут согласовывать дополнительные условия, отражающие специфику выполнения работ и особенно взаимоотношений сторон.

## II. Обязанности и права «Исполнителя»

7. «Исполнитель» обязан выполнять работы в сроки, определенные графиком, а также устранять неисправности техники по заявкам «Заказчика».

8. «Исполнитель» выполняет работы в объеме, установленном эксплуатационной и ремонтной документацией «Изготовителя». Кроме того, «Исполнитель» выполняет электромонтажные работы в кабинетах вычислительной техники в соответствии с 4 настоящего договора.

9. «Исполнитель» обеспечивает своевременный выезд специалистов к «Заказчику» для выполнения работ, включенных в график, сообщая «Заказчику» не позднее чем за 5 дней до начала работ состав командируемых специалистов для бронирования мест в гостинице или выделения мест в общежитии.

10. Для устранения выявленных в период эксплуатации техники неисправностей «Исполнитель» обязан прибыть к «Заказчику» в следующие сроки с момента получения его заявки (не считая времени, необходимого для проезда):

в течение 24 часов — по заявке одногородного «Заказчика»;

в течение 5 дней — по заявке иногороднего «Заказчика».

Исполнитель осуществляет восстановление работоспособности техники не позднее следующего дня после прибытия к «Заказчику». Если для выполнения работ требуется получение запасных частей, сроки ремонта удлиняются не более чем до 3 дней — с момента прибытия к одномуродному «Заказчику»;

до 5 дней — с момента прибытия к иногороднему «Заказчику».

Стороны могут во взаимной договоренности

установить иные сроки прибытия «Исполнителя» к «Заказчику» и устранения неисправностей.

11. «Исполнитель» обязан по получении от «Заказчика» сообщения о невыполнении или некачественном выполнении работ (п. 17 настоящего Типового договора) устранить за свой счет отмеченные недостатки в кратчайший технически возможный срок, согласованный сторонами, но не более 5 дней после получения сообщения.

12. В случае необеспечения «Заказчиком» к началу или в процессе выполнения работ соответствия кабинетов вычислительной техники требованиям эксплуатационной документации «Изготовителя», правилам техники безопасности и пожарной безопасности «Исполнитель» имеет право перенести срок выполнения работ с внесением согласованных с «Заказчиком» изменений в график и взыскать с «Заказчика» штраф и убытки, причиненные неисполнением либо ненадлежащим исполнением договорных обязательств.

13. Устранение неисправностей техники, возникших вследствие несоблюдения «Заказчиком» правил ее эксплуатации в гарантийный период, производится за счет «Заказчика». Несоблюдение правил эксплуатации техники отражается в актах, составляемых «Исполнителем» с участием «Заказчика». О всех случаях нарушения правил эксплуатации «Исполнитель» уведомляет «Изготовителя».

14. «Исполнитель» инструктирует представитель «Заказчика» по вопросам соблюдения правил эксплуатации техники.

## III. Обязанности и права «Заказчика»

15. «Заказчик» обязан:

а) обеспечить к началу проведения работ соответствие кабинетов вычислительной техники требованиям эксплуатационной документации «Изготовителя», правилам техники безопасности и пожарной безопасности;

б) распаковать и расконсервировать поступившую технику, проверить ее комплектность и наличие пломб «Изготовителя», установить технику с участием «Изготовителя» без подключения к энергосистемам;

в) представлять по заявкам «Исполнителя» на время выполнения работ рабочие места для специалистов «Исполнителя»;

г) обеспечить по заявке «Исполнителя» проживание его специалистов в гостинице или общежитии на весь срок выполнения работ.

16. В случае обнаружения несоответствия качества или комплектности техники или ее комплектующих изделий, узлов и деталей, поступивших к «Заказчику» от «Изготовителя», требованиям нормативно-технической документации или договорам на поставку «Заказчик» при участии «Исполнителя» обязан составить акт в соответствии с Инструкцией о порядке приемки продукции производственно-технического назначения и товаров народного потребления по качеству, утвержденной Государственным арбитражем СССР.

17. В случае невыполнения или некачественного выполнения «Исполнителем» работ «Заказчик» сообщает ему об этом в 2-дневный срок после обнаружения недостатков для принятия необходимых мер.

<sup>5</sup> В дальнейшем — «Изготовитель».

#### IV. Цена и порядок расчетов

18. Работы, выполненные «Исполнителем» по вводу в эксплуатацию, техническому обслуживанию и ремонту техники, оплачиваются «Заказчиком» по ценам \_\_\_\_\_

(номер и дата

\_\_\_\_\_ утверждения прейскуранта, протокола согласования

или другого документа)

19. Работы по вводу техники в эксплуатацию, а также по устранению неисправностей (ремонту) техники в послегарантийный период эксплуатации и при несоблюдении «Заказчиком» правил эксплуатации техники в гарантийный период оплачиваются «Заказчиком» на основании двусторонних актов сдачи-приемки выполненных работ, составляемых «Исполнителем» совместно с «Заказчиком» по окончании работ.

20. Оплата выполненных «Исполнителем» работ по техническому обслуживанию техники производится с «Заказчиком» ежемесячно равными долями в размере \_\_\_\_\_ руб.

(сумма прописью)

на основании двусторонних актов сдачи-приемки выполненных работ, составляемых «Исполнителем» совместно с «Заказчиком» ежемесячно. Стоимость технического обслуживания техники составляет за год \_\_\_\_\_ руб.

(сумма прописью)

21. Расчеты за выполнение работы производятся путем \_\_\_\_\_

(указывается принятая форма расчетов)

22. Расчеты за выполненные «Исполнителем» работы по техническому обслуживанию и ремонту техники в гарантийный период ее эксплуатации, при условии соблюдения «Заказчиком» установленных правил эксплуатации, производятся непосредственно между «Изготовителем» и «Исполнителем».

23. Общая сумма договора составляет \_\_\_\_\_

(сумма

\_\_\_\_\_ руб.

(прописью)

#### V. Дополнительные условия

24. Отдел (управление) народного образования прилагает к договору перечень школ, использующих вычислительную технику в учебном процессе.

В соответствии с частью первой статьи 38 Основ гражданского законодательства Союза ССР и союзных республик на школы возлагается исполнение обязательств, предусмотренных в пунктах \_\_\_\_\_

\_\_\_\_\_ настоящего Типового до-

говора. (перечислить пункты)

25. \_\_\_\_\_

#### VI. Ответственность сторон

26. «Исполнитель» уплачивает «Заказчику»:

а) за нарушение сроков выполнения работ, уста-

новленных графиком, а также за неустранение неисправностей техники в сроки, установленные п. 10 и 11 настоящего Типового договора, — штраф в размере 25 руб. за каждый день просрочки, но не более 500 руб.;

б) за невыполнение или некачественное выполнение работ — штраф в размере 50 руб. за каждый случай.

27. За ненадлежащее исполнение обязательств, предусмотренных подпунктами «а», «в» п. 15 настоящего договора, повлекшее за собой нарушение «Исполнителем» графика, «Заказчик» уплачивает «Исполнителю» штраф в размере 25 руб. за каждый случай.

28. За каждый необоснованный полный или частичный отказ от оплаты выполненных «Исполнителем» работ «Заказчик» уплачивает «Исполнителю» штраф в размере 5 процентов суммы, от уплаты которой он отказался. При несвоевременной оплате выполненных работ «Заказчик» уплачивает «Исполнителю» пеню в размере 0,04 процента суммы просроченного платежа за каждый день просрочки.

29. За просрочку заключения договора или уклонение от заключения его виновная сторона уплачивает другой стороне штраф в размере 10 руб. за каждый день просрочки, но не более 100 руб.

30. Независимо от уплаты штрафа (пени) сторона, нарушившая условия договора, возмещает другой стороне причиненные в результате этого убытки в части, не покрытой штрафом (пеней).

31. Уплата штрафа (пени) за нарушение условий договора, а также возмещение причиненных убытков не освобождают стороны от выполнения обязательств.

#### VII. Порядок разрешения споров

32. Споры технического характера разрешаются вышестоящими организациями сторон.

33. Споры, возникающие при исполнении договора, разрешаются в установленном законом порядке.

34. В случае досрочного расторжения договора производятся взаиморасчеты по фактическим расходам, произведенным сторонами на момент прекращения договорных обязательств.

#### VIII. Срок действия договора и юридические адреса сторон

35. Срок действия договора устанавливается с « » \_\_\_\_\_ 19 г. по « » \_\_\_\_\_ 19 г. с ежегодным уточнением перечня техники, графика и при необходимости других условий.

36. Договор считается продленным на следующий год, если ни одна из сторон не заявит о расторжении договора не менее чем за два месяца до истечения срока его действия.

37. «Заказчик» имеет право расторгнуть заключенный договор при превышении «Исполнителем» цены за выполненные работы.

38. По вопросам, связанным с выполнением обязательств по настоящему договору, стороны выделяют ответственных представителей от «Исполнителя»

(должность, фамилия, имя, отчество, служеб-

ный телефон)

от «Заказчика» \_\_\_\_\_

(должность, фамилия, имя, отчество, \_\_\_\_\_)

служебный телефон) \_\_\_\_\_

39. «Исполнитель»

код по общесоюзному классификатору \_\_\_\_\_

наименование, почтовый адрес, индекс, \_\_\_\_\_

телетайп \_\_\_\_\_

телефон \_\_\_\_\_

расчетный счет \_\_\_\_\_

(м. печати, подпись) \_\_\_\_\_

40. «Заказчик» \_\_\_\_\_

(наименование, ведомственная принадлежность, \_\_\_\_\_)

почтовый адрес, индекс) \_\_\_\_\_

телетайп \_\_\_\_\_

телефон \_\_\_\_\_

расчетный счет \_\_\_\_\_

(м. печати, подпись) \_\_\_\_\_

ПРИЛОЖЕНИЕ № 1  
К ДОГОВОРУ НА \_\_\_\_\_ ГОД

### Перечень техники

Наименование, номер (код) техники	Количество техники	Наименование, номер (код) выполняемых работ	Цена на услуги (на единицу техники)	Наименование документа и органа, утвердившего цену. Дата утверждения цены	Примечание

«Исполнитель» \_\_\_\_\_

«Заказчик» \_\_\_\_\_

ПРИЛОЖЕНИЕ № 2  
К ДОГОВОРУ НА \_\_\_\_\_ ГОД

Техника \_\_\_\_\_

(наименование, номер, код)

### График выполнения работ

№ п/п	Наименование, номер (код) выполняемых работ	Ответственный исполнитель	Сроки выполнения работ		Отчетный документ о выполнении работ	Примечание
			Начало	Окончание		

«Исполнитель» \_\_\_\_\_

«Заказчик» \_\_\_\_\_

Издательство «Педагогика» выпустило книгу Е. И. Машбица «Психолого-педагогические проблемы компьютеризации обучения».

Наиболее характерной проблемой для современного этапа компьютеризации обучения является разработка эффективных обучающих систем. Качество обучающих программ в большой степени определяется тем, насколько в них учтены особенности учебной деятельности, насколько они соответствуют психолого-педагогической теории обучения. К чему может привести игнорирование этого соответствия, показывает опыт компьютеризации обучения в развитых капиталистических странах, на ошибках которых лучше учиться, чем повторять их.

Анализу зарубежного опыта применения компьютера в учебном процессе, обсуждению взаимосвязи трех аспектов компьютеризации обучения — разработки психолого-педагогической теории обучения, создания технологии компьютерного обучения и проектирования обучающих систем — и посвящена эта книга.

Ю. МАТИЯСЕВИЧ

## Задачи первых ленинградских олимпиад по информатике

В 1985/86 учебном году после введения в школьную программу нового предмета в Ленинграде была проведена первая общегородская олимпиада по информатике и вычислительной технике. Инициатором выступил Ленинградский областной совет НТО, в составе которого в настоящее время работает Комитет по проблемам образования в области информатики, вычислительной техники и автоматизации.

Хотя проведение олимпиады и было связано с появлением еще одного школьного предмета, оргкомитет сразу принял решение, что олимпиадные задания не будут привязаны к учебному пособию под редакцией А. П. Ершова и В. М. Монахова. Это объяснялось невозможностью провести полноценное соревнование на основе того материала, который пройден всего за полгода. Было решено, что первая олимпиада будет ориентирована на учащихся школ и классов с физико-математическим уклоном. В связи с тем, что школьный тур проводился в разных школах и в разное время, были подготовлены два варианта (задачи 1, 3, 5 или 2, 4, 6, приведенные в конце статьи). После завершения соревнований в школах условия всех шести задач были опубликованы в ленинградской молодежной газете «Смена», с тем, чтобы учащиеся неспециализированных школ смогли принять участие в заочном отборе. Таких желающих, однако, оказалось менее двух десятков.

В 1986/87 учебном году соревнования на школьном уровне устраивались по усмотрению учителей, а затем проводился районный тур. По итогам предыдущего года пять лучших школ выступали на правах городских районов, но их учащиеся решали более трудные варианты Б задач 10—12.

Заключительный городской тур проводился оба раза по одинаковой схеме. Он состоял из двух этапов, разделенных двухнедельным перерывом. На первом этапе участники писали программы (задачи 7—9 и 13—15 соответственно), а на втором этапе лучшие из них отлаживали свои программы в дисплейных залах ЛГУ. Те, кто успевал это сделать, могли приступать к дополнительным задачам, решение которых требовало небольшой модификации первоначальных программ.

На всех этапах участники решали одни и те же задачи, но итоги проводились по IX и X классам.

На протяжении двух лет в оргкомитете шли споры о том, что такое олимпиадная задача по информатике. Существует устойчивое представление об олимпиадной задаче по математике, сложившееся за более чем полувековую историю проведения олимпиад. С информатикой дело обстоит сложнее уже потому, что еще нет единого мнения на информатику как науку и тем более как школьную дисциплину. Должна ли олимпиада по информатике отличаться от олимпиад по программированию и если должна, то чем? В какой мере в задачах по информатике могут присутствовать элементы олимпиадных задач по математике? Что и в какой степени ценить в решении?

Одно из первых положений, принятых оргкомитетом, состояло в том, что решение задачи должно быть доведено до отлаженной программы. Реальный выход на вычислительную технику удавалось организовать только для 60—70 участников второго этапа городского тура. Поскольку в разных школах изучаются разные языки, разрешалось писать программы на Алголе-60, Алголе-68,

Бейсике, Паскале, ПЛ/1, Рапире и Фортране. Разнообразие языков ограничивало круг возможных задач, ставило участников в неравные условия, затрудняло работу жюри. Представляется разумным перейти в дальнейшем на язык учебного пособия.

Отлаженные программы жюри проверяло на подобранной серии тестов. Это потребовало стандартизировать ввод-вывод, и вместе с условиями задач на городском туре участникам были даны единые (для каждого языка свои) начала и окончания программ.

На стадиях без выхода на реальные ЭВМ, правила были менее жесткие. Допускались и другие языки, в том числе язык из учебного пособия, участники самостоятельно выбирали форматы ввода-вывода.

Как показал опыт, проверять работы по информатике сложнее, чем математические, поэтому участники были обязаны сопровождать программы описаниями формы представления исходных данных, использованного алгоритма, особенностей реализации алгоритма в виде программы, формы выдачи результата. Эти требования стали камнем преткновения для многих. Оказалось, что некоторым легче выразить свои мысли на языке программирования, чем на родном языке.

При проверке работы на первое место ставилась *правильность*. Многие участники неверно понимали условия задач. Например, в задаче 2 давали ответ «да» при  $A=B$ , а в задаче 3 забывали про случаи  $K=0$  и  $K=1\ 000\ 000$ , в задаче 4 считали, что координаты вершин перечислены в порядке обхода контура, или даже полагали, что стороны прямоугольника параллельны осям, в задаче 13 требовали задания отрезков в порядке возрастания их левых координат.

На втором месте стояла *эффективность* программы. Например, в задаче 6 не засчитывалось «очевидное» решение с шестью вложенными циклами: если пять цифр уже известны, то шестая цифра «счастливого» билета определяется однозначно. Аналогично в задаче 1 ожидалось решение с тремя циклами, в задаче 11А — с одним. В некоторых задачах требование эффективности было сформулировано в виде примечания. К сожалению, это было понятно не всем, например, в задаче 7 многие довольствовались двумя вложенными циклами, считая, что тело цикла можно выполнять  $K(K+1)/2$  раз.

Кроме правильности и эффективности, жюри принимало во внимание такую трудноформализуемую характеристику программ, как *ясность*. Конечно, здесь участники были в неравном положении из-за различий в выразительных возможностях языков.

Что касается оценки трудности задач, вы-

яснилось, что предположения жюри иногда расходились с фактическими результатами. Например, задача 7 считалась трудной, но ее успешно (за один просмотр массива) решило довольно много участников. С другой стороны, задачи 8 и 15 считались легкими, но приемлемые рисунки были получены единицами.

Решение задачи 11, считавшейся очень простой, сопровождалось трудностями из-за того, что многие учащиеся воспринимали компьютер как большой арифмометр — пытались вывести явную формулу для  $K$  и запрограммировать ее.

**Задача 1.** Даны положительные числа  $A_1, A_2, A_3, A_4, M$ . Найти и напечатать все четверки положительных целых чисел  $X_1, X_2, X_3, X_4$ , удовлетворяющие уравнению  $A_1X_1 + A_2X_2 + A_3X_3 + A_4X_4 = M$ .

**Задача 2.** Даны два положительных целых числа  $A$  и  $B$ . Напечатать слово «ДА» или «НЕТ» в соответствии с тем, можно ли получить десятичную запись числа  $A$  путем вычеркивания одной или более цифр числа  $B$ .

**Задача 3.** Дано неотрицательное целое число  $K$ , не превышающее миллиона. Напечатать фразу « $K$  ВОРОН» русскими словами. (Пример: если  $K=23$ , то должно быть напечатано «ДВАДЦАТЬ ТРИ ВОРОНЫ»; если  $K=3651$  — то «ТРИ ТЫСЯЧИ ШЕСТЬСОТ ПЯТЬДЕСЯТ ОДНА ВОРОНА».)

**Задача 4.** Даны числа  $X_1, Y_1; X_2, Y_2; X_3, Y_3$  — координаты трех каких-то вершин прямоугольника в прямоугольной системе координат. Вычислить и напечатать координаты четвертой вершины.

**Задача 5.** Соревнования по прыжкам в высоту проводились по следующим правилам. Если спортсмен сбивал планку, то он выбывал из дальнейшей борьбы, в противном случае — становился в конец очереди. Известно, что спортсменов было  $N$ ; сначала они прыгали в порядке своих номеров от 1 до  $N$ , между двумя последовательными неудачами каждый раз было ровно  $M$  удачных прыжков; последним планку сбил спортсмен с номером  $T$ . Напечатать номера прыгунов в том порядке, в котором они сбивали планку. (Пример: если  $N=10, M=3, T=8$ , то должны быть напечатаны числа 7, 1, 5, 10, 6, 3, 2, 4, 9, 8.)

**Задача 6.** Трамвайные билеты имеют номера из шести цифр от 000000 до 999999. Билет называется «счастливым», если сумма 1, 3, 5-й цифр равна сумме 2, 4 и 6-й цифр. Вычислить и напечатать количество «счастливых» билетов.

**Задача 7.** Дано положительное целое число  $K$  и  $K$  целых чисел  $A_1, \dots, A_K$ . Вычислить наибольшее возможное значение суммы



$A_M + A_{M+1} + \dots + A_N$ , где  $1 \leq M \leq N \leq K$ .

**Примечание.** Число  $K$  столь велико, что числа  $A_1, \dots, A_K$  занимают примерно одну пятую памяти, отводимой для хранения данных, а на выполнение  $K^2$  даже простейших операций не хватает времени.

**Модификация.** Дано положительное целое число  $K$  и  $K$  целых чисел  $A_1, \dots, A_K$ , сумма которых равна 0. Числа были написаны по кругу. Вычислить максимальное значение суммы стоящих подряд чисел.

**Задача 8.** Устройство вывода вашей ЭВМ печатает на странице  $N$  строк, состоящих из  $M$  символов. Расстояние между центрами соседних символов равно: по горизонтали —  $H$  единиц длины, по вертикали —  $V$  единиц. Начало координат находится в центре страницы. «Нарисовать» символом «звездочка» две окружности с центрами в точках с координатами  $X_1, Y_1; X_2, Y_2$  и радиусами соответственно  $R_1$  и  $R_2$ .

**Примечание.** Если окружности не помещаются целиком на странице, то «рисовать» надо только те их части, которые помещаются.

**Задача 9.** Из листа клетчатой бумаги удалили некоторые клетки. На сколько кусков распадется оставшаяся часть листа? (Пример: если из шахматной доски удалить все клетки одного цвета, то оставшаяся часть распадется на 32 куска.)

**Модификация.** То же, но перед удалением клеток лист склеили в цилиндр высотой  $N$ .

**Задача 10. А.** Имеется  $N$  карточек. На каждой стороне каждой карточки написано одно целое число. Известно, что каждое из чисел  $1, \dots, N$  встречается на карточках дважды. Требуется узнать, можно ли карточки выложить так, чтобы каждое из чисел  $1, \dots, N$  было на верхней стороне одной из карточек; если можно, то указать для каждой карточки, как ее класть.

**Б.** Имеется  $N$  тетраэдров. На каждой грани каждого тетраэдра написано одно целое число. Известно, что каждое из чисел  $1, \dots, N$  встречается на тетраэдрах четыре раза. Требуется узнать, можно ли поставить тетраэдры на стол так, чтобы каждое из чисел  $1, \dots, N$  встречалось на верхних гранях трижды; если можно, то указать для каждого тетраэдра, как его ставить.

**Задача 11. А.** На клетчатой бумаге нарисовали окружность целого радиуса  $R$  с центром на пересечении линий. Найти  $K$  — количество клеток, целиком лежащих внутри этой окружности. (Пример: если  $R=5$ , то  $K=60$ .)

**Б.** Трехмерное пространство разбито на кубики с ребром длиной 1. Сколько кубиков помещается в сфере радиуса  $R$ , центр ко-

торой находится в вершине одного из кубиков? (Пример: в сфере радиуса 3 помещается 56 кубиков.)

**Задача 12. А.** Дана конечная последовательность, состоящая из левых и правых круглых скобок. Узнать, можно ли добавить в нее цифры и знаки арифметических действий так, чтобы получилось правильное арифметическое выражение.

**Б.** То же, но последовательность состоит из левых и правых круглых и квадратных скобок.

**Задача 13.** На прямой окрасили  $N$  отрезков. Известны координата  $L_i$  левого конца и координата  $R_i$  правого конца  $i$ -го отрезка для  $i=1, \dots, N$ . Найти сумму длин всех окрашенных частей прямой.

**Примечание.** Число  $N$  столь велико, что на выполнение  $N^2$  даже простейших действий не хватает времени.

**Модификация.** На окружности окрасили  $N$  дуг. Известны угловая координата  $L_i$  начала и  $R_i$  конца  $i$ -й дуги (от начала к концу двигались, закрашивая дугу, против часовой стрелки). Какая доля окружности окрашена?

**Задача 14.** Дана последовательность из  $N$  целых чисел, среди которых нет двух одинаковых. Требуется вычеркнуть минимально возможное количество чисел так, чтобы оставшиеся шли в порядке возрастания. На печать следует выдать  $K$  — количество оставшихся чисел и сами числа в порядке их следования.

**Примечание.** То же, что к задаче 13.

**Модификация.** Даны  $N$  положительных целых чисел, которые не делятся ни на какие числа, кроме 2 и 3. Требуется выкинуть минимально возможное количество чисел так, чтобы из любых двух оставшихся одно делилось на другое.

**Задача 15.** «Нарисовать» при помощи символа «звездочка» периодический узор в форме пчелиных сот, состоящий из одинаковых шестиугольников, максимально близких к правильным. Горизонтальные стороны шестиугольников должны быть образованы  $2W + 1$  звездочками, узор должен занимать  $L$  строк длиной 78 символов каждая (печатающее устройство может печатать более 78 символов на одной строке). Отношение расстояния по вертикали между соседними звездочками к расстоянию по горизонтали между соседними звездочками равно  $H$ .

**Модификация.** Будем считать, что соты рисовали не звездочкой, а символом «восклицательный знак» и полученная распечатка — обои, которыми надо оклеить стену шириной  $M$  символов и высотой  $N$  строк. Каково минимальное значение  $L$ , которого вам для этого хватит?

Г. ГОВОРУНОВ, М. ГУЩИНА

И. ИЛЬЧЕНКО,  
канд. мед. наук

Н. ПОРТЯНЕНКО,  
канд. мед. наук

## Среднее медицинское училище и компьютер

Современная дидактика ставит на первый план развитие творческого мышления учащихся. Особую значимость это приобретает в медицинских училищах.

Работа фельдшера и акушерки в условиях фельдшерско-акушерского пункта приближается к работе врача. От качества умственного профессионального труда зависит стабильность, успешность, быстрота и качество выполняемых задач, которые постоянно возникают перед медиками. Требования к специалистам растут непрерывно, а методы обучения остаются на прежнем уровне. Поэтому для существенного повышения качества подготовки специалистов нужны принципиально новые подходы.

Подготовка специалистов любого профиля включает в себя несколько этапов. Это в первую очередь описание объекта, дальнейшее распознавание его и, наконец, выработка умений, практических навыков.

Подготовка фельдшеров и акушерок также включает в себя все эти процессы: фельдшер должен знать описание болезней, уметь быстро и безошибочно распознавать заболевание и оказать больному эффективную помощь. В процессе обучения необходимо сформировать у учащихся клиническое мышление, научить их работать сначала в имитированных условиях, а затем уже с реальными больными.

Для этой цели могут быть использованы дидактические (имитационные) игры, основная цель которых заключается в создании для учащихся реальной модели их дальнейшей профессиональной деятельности.

Последние два года в Ростовском областном базовом медицинском училище все шире внедряются методы игрового обучения, в которых первостепенную роль играют эмоции

учащихся. Целесообразность имитационных игр очевидна. Ведь работа с настоящими больными должна быть завершающей степенью обучения, ибо больных травмирует, когда они превращаются в некое учебное пособие. При такой постановке ежедневных практических занятий с группами больных возникает целый ряд этических, психологических и деонтологических проблем. Кроме этого, преподаватели-клиницисты довольно часто сталкиваются на занятиях с отсутствием так называемых «тематических» больных. Применение в процессе обучения имитационных или дидактических игр оптимально решает эти вопросы. Главное назначение имитационных игр — максимально уменьшить для учащихся степень новизны и неожиданности в различных клинических ситуациях.

В своей работе мы взяли за основу карточный вариант клинических игр, разработанный в Новосибирском медицинском институте, на кафедре оптимизации высшего медицинского образования, профессором Л. Б. Наумовым. Карточный вариант оказался простым, удобным, применимым в любой обстановке. Преподавателями училища были рассмотрены и внедрены в учебный процесс карточные имитационные игры по курсам хирургии, терапии, педиатрии, акушерства и ЛОР болезням, включающие в себя самые различные ситуации: «фельдшер с одним больным», «с несколькими больными», «консилиум специалистов», «бригада скорой помощи». Основные игровые ситуации разрабатывались для «острых» течений болезней, так как именно в «острой» ситуации возникает ни с чем не сравнимое ощущение реальной ответственности за жизнь и здоровье больного. При создании имитацион-

ных игр мы использовали те разделы учебного курса, которые обеспечивают дифференциальную диагностику заболеваний и оказание первой помощи. Игры могут быть использованы лишь на тех этапах обучения, когда сформировалось определенное клиническое мышление, основанное на полученных знаниях, т. е. на V—VII семестрах и всех этапах преддипломной практики.

Разрабатывая клинические игры по разным дисциплинам, мы постепенно создаем банк игр. Составлять игру нужно сразу на высоком уровне, помня, что создать ее трудно. Необходимо знать теоретические основы разработки, которые описаны в монографии Л. Б. Наумова «Деловые игры в медицине».

Что же представляет собой синдромный принцип мышления, на основании которого мы создавали имитационные или дидактические игры? Л. Б. Наумовым определены понятия как синдрома, так и синдромного метода мышления, который взят за основу в нашей работе: «Синдромный принцип диагностики — это распознавание и разграничение болезней, на основу которого положены мыслительные операции только с тем кругом патологических процессов, которые проявляются ЕДИНЫМ ВЕДУЩИМ СИНДРОМОМ независимо от их принадлежности к разным нозологическим единицам, группировкам болезней, даже если они поражают разные органы и системы». Сравнивая традиционный принцип мышления с синдромным, легко убедиться, что на всех этапах мышления при нозологическом принципе диагностики мы имеем дело с огромным объемом информации и эксплуатируем в основном свою память, а при синдромном — с единичными элементами информации и оттачиваем экономное результативное логическое мышление. Причем разные синдромы столь явно отличаются друг от друга, что спутать их невозможно («боль в груди», «боль в животе», «тахикардия», «гиперемия» и т. д.). И в этом — важнейшее преимущество синдромного принципа диагностики.

Однако при всех описанных выше положительных качествах этого варианта дидактических игр мы вынуждены признать и некоторое их несовершенство: невозможность применить их в качестве тренажеров при подготовке учащихся к итоговому контролю. С помощью карточного варианта не удается также индивидуализировать подход к каждому студенту, принимающему участие в игре. Однако эти недостатки могут быть сведены на нет при использовании в учебном процессе электронно-вычислительных машин.

Математическая модель, реализованная на ЭВМ, позволяет не только имитировать состояние больного человека, но и дает воз-

можность учащимся вести диалог с «больным», постоянно наблюдать его состояние, проводить лабораторные и инструментальные исследования, «оказывать» экстренную медицинскую помощь.

Повышению наглядности и эффективности компьютерного варианта игры содействует работа модели в реальном, замедленном или убыстренном времени, что обеспечивается функционированием модели под управлением программируемого таймера. Его применение дает возможность имитации различных (от вялотекущих до молниеносных) форм болезни путем изменения масштаба времени.

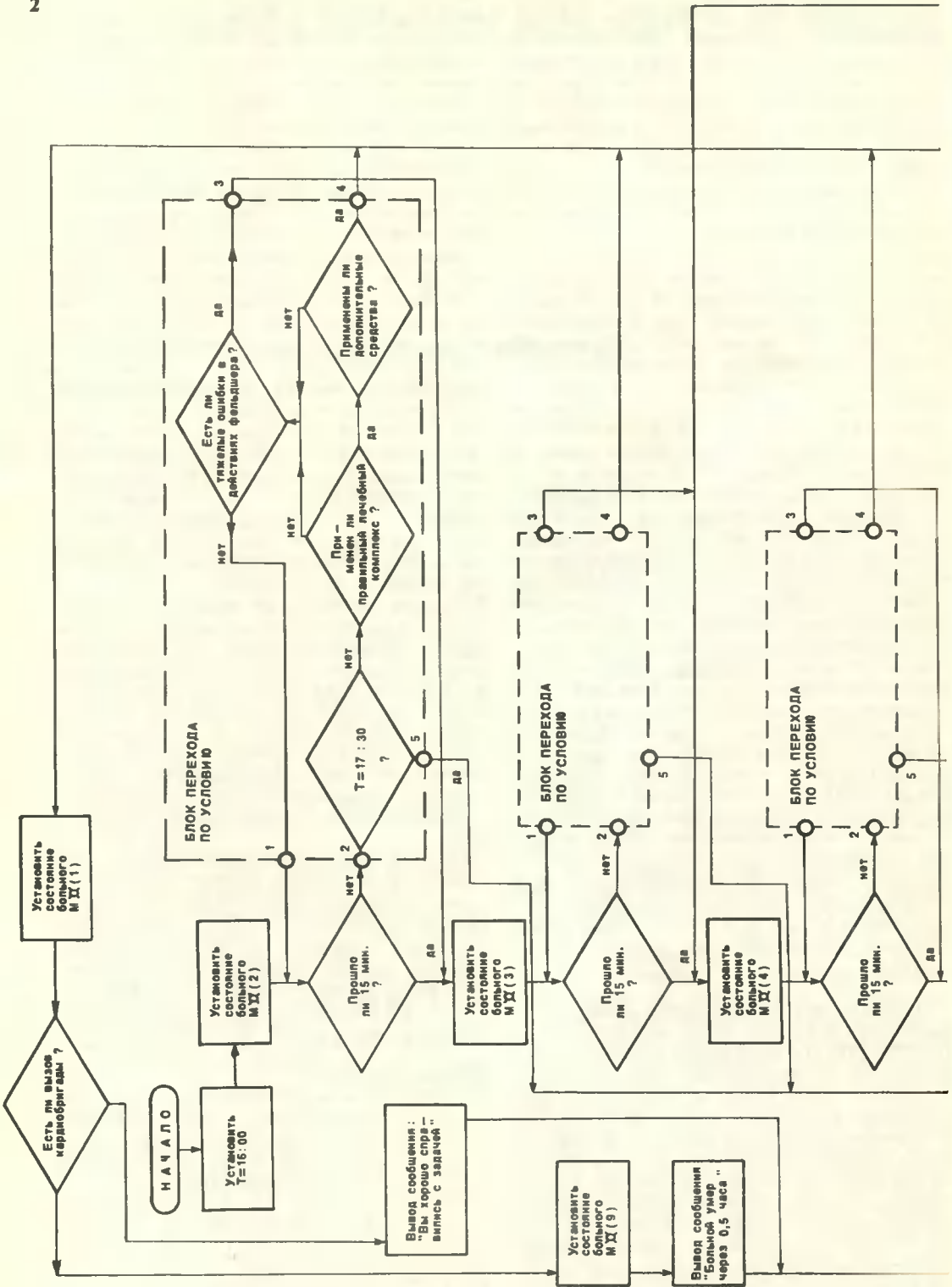
В Ростовском областном базовом медицинском лицее в начале 1986 г. был реализован компьютерный вариант имитационной игры «Боль в груди», прошедший апробацию на практических занятиях фельдшерского отделения училища и на областном педагогическом совещании профессорско-преподавательского состава РОДНМИ, преподавателей медицинских училищ области и методистов базовых училищ Российской Федерации. Описываемая игра сходна по принципу реализации со множеством уже разработанных в настоящее время динамических деловых игр, где время является одним из важнейших факторов, влияющих на моделируемый процесс. Тем не менее авторам не удалось найти аналогий нижеприведенной реализации имитационной модели больного человека или же описания практического применения подобных моделей. Отличительной особенностью явилось применение персональной ЭВМ «Искра-226», которая обладает компактностью и высокой надежностью, уже отлично зарекомендовала себя в системах автоматизированной диспансеризации населения.

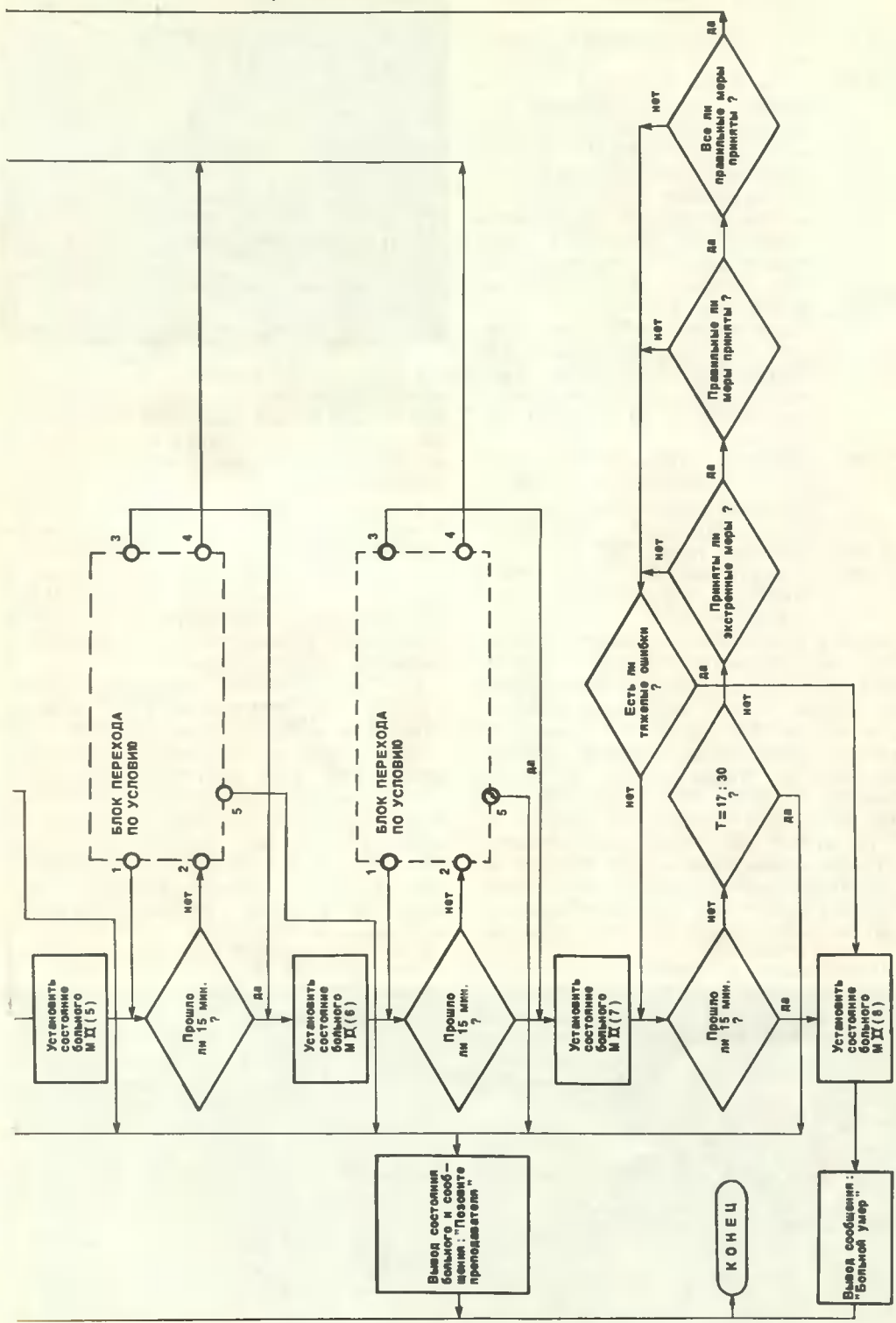
Основная концепция компьютерной реализации на языке Бейсик состоит в следующем.

Весь процесс болезни разбивается на части, каждая представляет собой определенное состояние больного с точки зрения наблюдающего за ним фельдшера. Для описываемой игры такие состояния изображаются в виде определенного наполнения соответствующих переменных символьного массива  $M \text{ } \text{X} \text{ } (1)$  (см. рис. 1) и принимают следующие значения:

$M \text{ } \text{X} \text{ } (1)$  — «Боли значительно ослабли. Самочувствие хорошее. Сознание ясное. При аускультации сердца тоны слегка приглушены. Ритм сердца правильный. Пульс 75 в 1 мин хорошего наполнения. АД 110/70 мм рт. ст. Со стороны других систем и органов без особенностей».

$M \text{ } \text{X} \text{ } (2)$  — «Вызов «скорой помощи» в 16<sup>00</sup>. Больной 40 лет, лежит в постели.





Установить состояние больного И Д (5)

Прошло ли 15 мин.

Установить состояние больного И Д (6)

Прошло ли 15 мин.

Установить состояние больного И Д (7)

Прошло ли 15 мин.

Установить состояние больного И Д (8)

Выход сообщения: "Больной умер"

КОНЕЦ

Выход сообщения: "Больной умер"

БЛОК ПЕРЕХОДА ПО УСЛОВИЮ

БЛОК ПЕРЕХОДА ПО УСЛОВИЮ

Есть ли тяжелые ошибки ?

Т = 17 : 30 ?

Прошло ли 15 мин.

Приняты ли экстренные меры ?

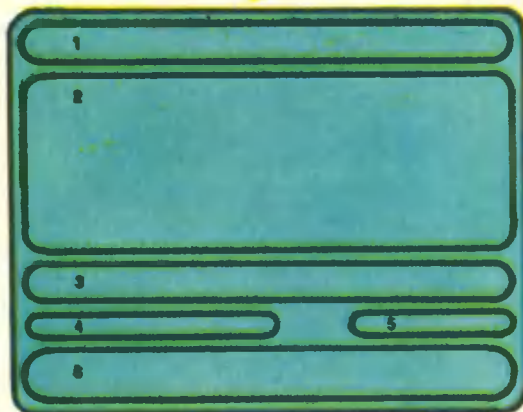
Правильные ли меры приняты ?

Все ли правильные меры приняты ?

Бледен. Беспокоен из-за резких болей в груди».

- М ♂ (3) — «Боль в груди. Одышка. Цианоз губ».
- М ♂ (4) — «Боль в области сердца. Сильная одышка. На лице больного холодный липкий пот. Головокружение. Пульс 90 в 1 мин слабого наполнения. АД 90/50 мм рт.ст.».
- М ♂ (5) — «Значительные боли в области сердца. Одышка. Акроцианоз. Холодный пот. Пульс 95 в 1 мин. Акцент второго тона над легочной артерией. АД 60/30 мм рт. ст.».
- М ♂ (6) — «Боли не стихают. Число дыханий 27 в 1 мин. Акроцианоз. Холодный пот. Пульс 50 в 1 мин. Акцент второго тона над легочной артерией. АД 60/30 мм рт. ст.».
- 92 М ♂ (7) — «Больной потерял сознание. Зрачки расширены. Пульс и артериальное давление не определяются. Дыхание отсутствует».
- М ♂ (8) — «Больной умер».
- М ♂ (9) — «Смерть больного через 0,5 ч после отъезда фельдшера «скорой помощи».

Значения программного таймера, работающего с коэффициентом пересчета 6:1, вызывают прерывания в работе остальных программ каждые 15 мин машинного времени, т. е. в течение 1,5 ч реального времени у больного, лишённого медицинской помощи, развиваются все этапы болезни, вплоть до летального исхода, что сопровождается сменой состояний больного во времени от М ♂ (2) до М ♂ (8). Упрощенный алгоритм функционирования этой модели показан на рис. 1. Информация выводится на дисплей. Все рабочее поле экрана функционально разделено на шесть полей, представленных на рис. 2.



1 — поле служебного текста. В него выводятся тексты, играющие вспомогательную роль, например такие: «Вы можете получить ответы на следующие вопросы:», «Какой из указанных лечебных комплексов вы примените?»

2 — поле вопросов, состоящее из 13 строк. В них располагаются вопросы, которые может задавать фельдшер «больному» в процессе сбора анамнеза. В этих строках в процессе проведения лечения или исследования «больного» появляются соответственно наименования лекарственных комплексов или методов исследования.

3 — поле ответов. Представляет собой 3 строки, куда ЭВМ выводит ответ от имени «больного». В этом же поле появляются системные сообщения об ошибках, допущенных фельдшером, и результаты исследований.

4 — служебное поле состояния «больного». В моменты смены состояний «больного» появляется мигающая надпись «состояние больного улучшилось» или «состояние больного ухудшилось», сопровождающаяся прерывистым звуковым сигналом, привлекающим внимание учащегося.

5 — поле таймера. В это поле постоянно

**ВЫ МОЖЕТЕ ПОЛУЧИТЬ ОТВЕТЫ НА СЛЕДУЮЩИЕ ВОПРОСЫ:**

- ЛОКАЛИЗАЦИЯ БОЛИ?
- СИЛА БОЛИ?
- ХАРАКТЕР БОЛИ?
- ДЛИТЕЛЬНОСТЬ БОЛИ?
- ИРРАДИАЦИЯ БОЛИ?
- СВЯЗАНА ЛИ БОЛЬ С ЧЕМ-ЛИБО?
- ЕСЛИ НЕТ ВОПРОСОВ

**БОЛИ ЛОКАЛИЗУЮТСЯ В НИЖНЕЙ ЧАСТИ ГРУДИНЫ**

**СОСТОЯНИЕ БОЛЬНОГО УХУДШИЛОСЬ**

**ВРЕМЯ 16:30:20**

**БОЛЬ В ОБЛАСТИ СЕРДЦА, СИЛЬНАЯ ОДЫШКА. НА ЛИЦЕ БОЛЬНОГО ХОЛОДНЫЙ ЛИПКИЙ ПОТ. ГОЛОВОКРУЖЕНИЕ. ПУЛЬС 90 В 1 МИН СЛАБОГО НАПОЛНЕНИЯ. АД. 90/50 ММ РТ. СТ.**

**ВЫ МОЖЕТЕ:**

**УЗНАТЬ ЖАЛОБЫ БОЛЬНОГО  
УЗНАТЬ О ПРИЕМЕ ЛЕКАРСТВ ДО ВАШЕГО ПРИЕЗДА  
ВЫЯСНИТЬ АНАМНЕЗ БОЛЕЗНИ  
ВЫЯСНИТЬ АНАМНЕЗ ЖИЗНИ  
ПРОВЕСТИ ОБЪЕКТ. ИССЛЕДОВАНИЕ:**

- А) ПО ОБЩИМ СИМПТОМАМ**
- Б) СИСТЕМЫ ОРГАНОВ ДЫХАНИЯ**
- В) СЕРД.-СОСУДИСТОЙ СИСТЕМЫ**
- Г) ДРУГИХ ОРГАНОВ**

**ПОЛУЧИТЬ ЛАБОРАТОРНЫЕ ДАННЫЕ  
ПРОВЕСТИ ИНСТРУМЕНТ. ИССЛЕДОВАНИЕ (ТЕМПЕРАТУРА, А.Д., ЭКГ, ФКГ)  
ПОСТАВИТЬ ПРЕДПОЛОЖИТЕЛЬНЫЙ ДИАГНОЗ  
ПРОВЕСТИ ЛЕЧЕНИЕ**

**СОСТОЯНИЕ БОЛЬНОГО**

**ВРЕМЯ 16:00:00**

**ВЫЗОВ «СКОРОЙ ПОМОЩИ» В 16.00. БОЛЬНОЙ 40 ЛЕТ. ЛЕЖИТ В ПОСТЕЛИ.  
БЛЕДЕН. БЕСПОКОЕН ИЗ-ЗА РЕЗКИХ БОЛЕЙ В ГРУДИ.**

выводятся показания таймера. Фактически это поле является электронными часами с цифровым отсчетом времени.

б — поле состояния «больного». В это поле выводится содержимое переменных M X (X), т. е. описание ситуации или текущего состояния «больного».

Пример одного из текущих состояний экрана приведен на рис. 3. Все управление ЭВМ сводится к нажатию двух клавиш: «пробел» и «возврат каретки». Многократным нажатием клавиши «пробел» пользователь перемещает светящуюся стрелку, расположенную слева от вопросов, в поле 2, до ее совмещения с нужным вопросом, а после нажатия клавиши «возврат каретки» получает ответ на вопрос в поле 3 и соответствующую реакцию «больного» в полях 4 и 5. В данном варианте игры имеются четыре возможных окончания:

1. Больному оказана своевременная и правильная медицинская помощь.

2. Больному оказана медицинская помощь, но не в полном объеме из-за отсутствия у фельдшера времени на посещение больного.

3. Больной умирает в присутствии фельдшера «скорой помощи».

4. Больной умирает через 0,5 ч после отъезда фельдшера.

Во втором случае ЭВМ прекращает работу и предлагает учащемуся позвать преподавателя для оценки проведенной работы.

Игра начинается с вывода на экран инструкции следующего содержания:

«Уважаемый коллега!

Вам предлагается вариант имитационной игры на синдром боли в груди. Вы — фельдшер «скорой помощи»!

Помните! Максимальное время на обслуживание вызова с оказанием помощи и наблюдением за эффективностью мероприятий 1 ч

30 мин. Каждая минута машинного времени соответствует 6 мин реального времени! Тактика фельдшера в игре базируется на действующих инструкциях Министерства здравоохранения СССР. Цель Вашего присутствия у постели больного — оказать эффективную медицинскую помощь.

После того как Вам будет предложена исходная ситуация, Вы сможете провести традиционное исследование больного по схеме: жалобы, anamnesis morbi, anamnesis vitae, status praesens, status localis, дополнительные данные. От Вашей квалификации зависит кратчайший путь установления диагноза, а следовательно, и эффективная помощь. ЭВМ будет предлагать Вам варианты действий. Выбор варианта осуществляется многократным нажатием клавиши «пробел» до совмещения светящейся стрелки с выбранным вариантом и последующим нажатием клавиши. По окончании ознакомления с инструкцией нажмите клавишу и попрактикуйтесь в общении с ЭВМ в режиме «Тренажера».

После нажатия клавиши «возврат каретки» пользователь имеет возможность попрактиковаться в течение 3 мин в режиме «Тренажер», который позволяет учащемуся овладеть клавишами управления и узнать на практике о назначении отдельных полей экрана. В этом режиме ЭВМ отвечает в юмористическом тоне на различные вопросы пользователя. По окончании тренинга пользователь сообщает, что у него больше нет вопросов, и ЭВМ, пожелав удачи, переходит в режим имитационной игры, которая начинается с экрана, показанного на рис. 4.

Диалог с пользователем в течение игры ведется способом, описанным выше, а алгоритм смены состояний «больного» соответствует рис. 1. В данной статье приведен упрощенный вид алгоритма. Реальный же

алгоритм имеет бóльшую вложенность подпрограмм, позволяет анализировать текущее состояние больного. Например, «больной» не может отвечать на вопросы фельдшера, находясь в бессознательном состоянии. В этом случае в поле 3 выводится сообщение: «В таком состоянии задавать вопросы больному бессмысленно!»

Для расширения возможностей игры при дальнейшей ее модификации в программу включены три буферные переменные, в которых фиксируется не только номер текущего состояния больного, но и номера двух предыдущих состояний. Это в перспективе дает возможность более деятельного анализа прошедших состояний с целью запрета повторных действий до стороны фельдшера, что в свою очередь поможет полностью исключить «лечение» методом последовательного перебора предлагаемых лекарственных комплексов. В дальнейшем предполагается разработка программных сегментов, предназначенных для участия пользователя в других ролях, например фельдшера кардио- или реанимационной бригады, фельдшера приемного отделения. Этот метод дает возможность реализовать на персональных ЭВМ большие

имитационные игры с продолжением.

Кроме всего прочего, нельзя упускать из виду, что описанный подход к созданию имитационных игр дает возможность использовать эти же игры не только в качестве контролеров знаний, но и в качестве имитационных тренажеров, где каждый правильный или ошибочный шаг фельдшера будет сопровождаться кратким комментарием относительно того, чем хорошо или плохо то или иное действие фельдшера и в чем состоят его ошибки. Естественно, создание таких компьютерных игр требует неперемного участия двух специалистов: программиста и профессионала-медика.

По мере осуществления программы всеобщего компьютерного образования и по мере создания в средних медицинских учебных заведениях дисплейных классов компьютерные имитационные игры, подобные описанной выше, будут занимать все большее место в учебном процессе.

\* От редакции. По поводу программ заинтересованные лица могут обратиться по адресу: 344082, г. Ростов-на-Дону, ул. Пушкинская, д. 13, кв. 12. Говорунову Г. В.

**О. ИВАНОВА,**  
преподаватель Техникума государственного делопроизводства,  
Москва

## Опыт обучения работе на клавиатуре персональных ЭВМ

По мере развития вычислительной техники, ее удешевления, увеличения доступности для все большего числа работников аппарата управления, научных работников, студентов все острее ощущается необходимость научить неспециалистов работать с ЭВМ, пользоваться клавиатурой ввода информации.

Нередко можно видеть, как человек, работающий с современной, многофункциональной, быстродействующей машиной, вводит информацию одним-двумя пальцами, выскивая на клавиатуре ту или иную букву, знак. Эффективность использования машины в этом случае невелика. И причиной тому — неподготовленность пользователей к оперативной работе с ЭВМ. Отсутствие клавиатурной грамоты вызывает трудности общения с машиной, приводит к потерям времени.

Один из путей повышения культуры работы на персональных ЭВМ — обучение пользователей работе с клавиатурой.

Необходимо отметить, что расположение клавиш с буквами и цифрами на клавиатуре персональных ЭВМ (за исключением букв «ъ» и «ё») такое же, как и на пишущей машине, различие только в расположении знаков. Следовательно, обучить работе на клавиатуре дисплея — значит дополнительно дать возможность печатать на пишущей машине. Кроме того, необходимо использовать возможности компьютера не только как предмета изучения, но и как средства обучения.

О том, чтобы научить каждого желающего быстрому вводу информации (особенно текстовой) в ЭВМ, думают многие. Например, в Московском институте электронного машиностроения и Новосибирском государственном педагогическом институте разработаны машинные программы для ознакомления с клавиатурой ввода информации. И если программа МИЭМ носит чисто ознакомительный характер, т. е. на экране в определенной



последовательности предъявляются клавиши с пояснениями их функционального назначения, то программа, разработанная в Новосибирске, предназначена (как это указано в ее начале) для обучения печати десятипальцевым слепым методом — основным при обучении машинописи. Она построена по принципу самоучителя. Вся информация поступает с экрана компьютера. Курс обучения разбит на уроки, каждый из которых легко вызвать на экран для работы.

Курс интенсивного автоматизированного обучения машинописи предлагает В. Петрусинский\* на основе сугестокибернетического метода. Обучение проводится на клавиатуре пишущей машины или специальном сенсорном тренажере. Обучающая программа представляется с помощью зрительных и звуковых сигналов, которыми задаются инструкции по управлению состоянием, а также по системе работы на клавиатуре. Музыкальными сигналами фоновой музыки задается темп работы на клавиатуре. Основное при обучении этим методом — максимальная интенсификация учебного процесса путем так называемой информационной стимуляции. Интенсивный режим работы чередуется с активным отдыхом: зарядкой, гимнастикой для пальцев, ритмической гимнастикой.

Разработчики наиболее известных обучающих программ обратились к практике учебных заведений по подготовке машинисток и в основу своих программ положили один из существующих способов изучения клавиатуры — горизонтальный (по рядам), вертикальный (по зонам), горизонтально-вертикальный, по парам букв. Эти способы основаны на механистическом подходе к образованию новых навыков и опираются на принципы последовательного изучения клавиатуры и многократного повторения одного и того же движения для его освоения. Учащийся приобретает навык путем многочисленных проб и ошибок, без управляющего воздействия со стороны преподавателя. Это приводит к тому, что на изучение клавиатуры уходит много времени (до 100 учебных часов), и многие так и не овладевают навыком слепой работы.

Использование новой техники в условиях массового обучения требует организации учебного процесса на основе современных психологических представлений о формировании профессиональных знаний и умений, с учетом передовых достижений педагогики и в первую очередь дидактики.

Поиски таких подходов привели ряд иссле-

дователей\* к теориям А. Леонтьева, Б. Ломова, Н. Бернштейна, П. Гальперина, которые в настоящее время в различных областях обучения вызывают большой интерес и связаны с подходом к навыку как сознательно формируемому действию.

Согласно этому подходу, управление исполнением любого действия на основе сознательных образов является определяющей характеристикой человеческой деятельности. Вместе с тем в процессе овладения тем или иным действием мера участия осознания в регуляции исполнения уменьшается. Так, например, если для начинающей машинистки (оператора ЭВМ) для напечатания (набора) буквы требуется выполнять ряд осознанных действий (найти местоположение заданной буквы на клавиатуре, определить, каким пальцем и в каком направлении необходимо осуществить движение, и др.), то для опытного работника восприятие буквы как бы само вызывает необходимое движение.

Изменение осознания действия по мере овладения им позволило советским психологам сформулировать основной принцип превращения действия в навык: на начальном этапе формирования действия обучаемый должен осознавать и цель действия, и способ ее достижения, а на конечном этапе для получения конечного результата действия необходимо осознание лишь заданной цели. Исходя из этого следует, что на начальном этапе обучения работе на клавиатуре необходимо создать условия, при которых обучаемые сознательно отрабатывают умение определять местоположение заданной буквы на клавиатуре, моторную характеристику (т. е. каким пальцем и в каком направлении необходимо совершить движение) и совершать движение на ее основе. На заключительном этапе действия становятся неосознанными и выполняются автоматически.

Данный подход к овладению двигательными навыками, выделение четких последовательных стадий в их освоении, позволяет провести операциональное описание данного метода обучения, т. е. его технологизацию. Технологизация метода обучения в свою очередь позволяет разработать машинные программы для компьютерного обучения, полностью или частично передать функцию обучения самой машине.

Первая попытка создания машинных обучающих программ на основе такого подхода

\* Петрусинский В. Автоматизированные системы интенсивного обучения. М.: Высшая школа, 1987. С. 117—127.

\* Малов С., Елина И. Новое в методике обучения машинописи // Вопросы повышения качества преподавания стенографии, машинописи и делопроизводства: Материалы семинара, проведенного 27—28 августа 1981 г. в Москве.



была предпринята членами кружка «Техника работы на дисплее» средней школы № 57 Ленинского района Москвы Андреем Зязиным, Андреем Николаичиком, Михаилом Артемьевым и препод. О. Ивановой под техническим руководством зав. лабораторией компьютеризации школьного обучения НИИ школ А. Матвеева.

Учащиеся этой школы имеют возможность углубленно изучать программирование. И не только теоретически, но и практически в школьном кабинете информатики, оснащенном отечественными персональными компьютерами «Агат», БК-0010, а также японскими «Ямахами».

Высокий уровень математической подготовки, владение Бейсиком, возможность практической работы на ЭВМ позволили учащимся в свободное от уроков время разработать комплект из 18 программ по изучению клавиатуры персональных ЭВМ.

Отдельные программы были опробованы на занятиях кружка. С ноября 1986 г. по май 1987 г. было обучено 4 группы учащихся IX—X классов — всего 49 человек. На изучение клавиатуры дисплея затрачивалось не более 20 учебных часов. Замер скорости выполнения заданий по буквенной клавиатуре показал, что у учащихся сформировался устойчивый навык ввода информации со скоростью в среднем 65 уд./мин (при разбросе от 40 до 100 ударов).

Помимо обучения работе на клавиатуре дисплея, учащиеся знакомились с историей создания, развитием и устройством пишущих машин, их системами, техникой безопасности при работе с ними, учились правильно оформлять машинописные работы. Это позволило привлекать членов кружка для помощи администрации школы в выполнении отдельных машинописных работ, оформления заметок для школьной стенгазеты. Для учителей-предметников члены кружка вводили текстовую информацию в ЭВМ.

Опыт кружка «Техника работы на дисплее» подтвердил: владение клавиатурой персональных ЭВМ сокращает затраты времени на ввод информации в машину, прививает культуру работы с ней. Однако нерегулярность и необязательность посещения всех занятий не позволили добиться желаемого результата для всех учащихся.

Весь комплект программ был использован в марте 1987 г. на занятиях с группой преподавателей средней школы № 117 Черемушкинского района Москвы на базе методического кабинета ЭВМ НИИ школ.

Курс обучения работе на клавиатуре дисплея, состоящий из трех занятий по четыре учебных часа, был проведен с целью дальнейшего использования преподавателями современной вычислительной техники в школе при проведении уроков. Он рассчитан на групповое обучение (количество занимаю-

шихся зависит от числа рабочих мест). Объяснение нового материала, основной инструктаж дает преподаватель, выполнение практических заданий проводится с помощью ЭВМ. Время выполнения каждого задания — общее для всех, что позволяет преподавателю работать одновременно со всеми обучаемыми, вести целую группу.

Процесс изучения клавиатуры ЭВМ по новой методике\* включает: изучение буквенного состава зон: формирование навыков печати по зонам, побуквенной печати по маршруту движения, побуквенной печати по местоположению буквы, побуквенной печати по букве; изучение цифр и знаков.

Разучивание буквенного состава клавиатуры происходит под руководством преподавателя без клавиатуры ЭВМ, по определенной системе и с привлечением новых форм заданий. В зависимости от состава группы этот этап занимает от 40 до 60 мин. Определение местоположения букв и отработка печати по маршруту движения происходят с помощью обучающих машинных программ. Информация подается в виде специально разработанных схем с экрана ЭВМ, задания с текстовой информацией выполняются учащимися под контролем ЭВМ. Чередуя подачу информации с экрана и выполнение заданий с листа позволяют поддерживать высокую работоспособность учащихся, избежать утомления глаз, характерного при работе только с экрана.

В процессе обучения выявлены некоторые недостатки программ, связанные в основном с контролем скорости и качества выполнения заданий обучаемыми. Разработанные школьниками программы будут совершенствоваться с целью индивидуализации обучения, разгрузки преподавателя, унификации. Будут учтены замечания и пожелания обучаемых.

После семи часов обучения контрольный замер выполнения заданий по буквенной клавиатуре показал, что обучаемые вводят информацию слепым десятипальцевым способом со средней скоростью 40 уд./мин и средним числом ошибок 0,65 за 5 мин работы, т. е. был достигнут желаемый уровень качества работы. Что касается скорости, то она возрастает до заданного уровня (60 уд./мин) к концу изучения клавиатуры. В целом занятия прошли успешно и позволили учителям освоить работу на клавиатуре ЭВМ слепым десятипальцевым способом, и убедили в необходимости продолжения

подобных курсов обучения для учителей, о чем свидетельствуют анкеты, заполненные ими на последнем занятии.

Благодаря принципиально новому подходу к формированию двигательного навыка данная методика применительно к обучению работе на клавиатуре персональных ЭВМ с помощью машинных обучающих программ позволяет:

сократить сроки изучения клавиатуры до 12 ч, что дает возможность обучить при необходимости большое число пользователей;

обеспечить качество и оперативность работы пользователя персональных ЭВМ при вводе больших объемов текстовой информации;

обеспечить при необходимости слепой метод набора информации, что значительно повышает скорость ввода информации в машину и комфортность работы;

повысить интерес обучаемых к овладению новыми навыками за счет разнообразия форм и методов работы, использования игровых заданий.

*От редакции.* Более подробную информацию можно получить по тел.: 141-03-59 у Ивановой Ольги Владимировны



\* Подробнее о проведении занятий см.: Методические материалы по обучению пользователей персональных ЭВМ работе на клавиатуре с помощью программных тренажеров. М.: НИИ школ, 1987.

## «Агат» в нашей школе

Хочу поделиться впечатлениями об опыте преподавания информатики в экспериментальной школе № 625 АПН СССР, а также предложить ряд идей, которые, с моей точки зрения, могли бы усовершенствовать процесс освоения компьютеров в школе.

Школа № 625 имеет компьютерный класс (см.: Информатика и образование. 1987 № 1), укомплектованный машинами «Агат». Преподавание информатики в школе ведут научные сотрудники ВЦ АН СССР. Последние два года школьников IX—X классов обучали навыками работы с машиной, программированию на Бейсике и Рапире, некоторым вычислительным методам.

98 Об «Агате». Преимущество серийного «Агата» перед другими советскими ПЭВМ — наличие цветного дисплея. Возможность использования разных цветов значительно повышает интерес учащихся к машине и к информатике вообще. Главный недостаток «Агата» — его ненадежность. Это особенно относится к двум узлам — дисководу и клавиатуре.

Сбои при обращении к диску на наших машинах происходили катастрофически часто и иногда с необратимыми последствиями. Часть сбоев можно объяснить недостаточно аккуратным обращением школьников с дисками. Но это не единственная причина. Дисковод неустойчиво работает и с качественными дисками. Например, информация, записанная на одной машине, может не читаться на другой. Следует заметить, что ученики должны были интенсивно использовать дисковую память — они решали довольно сложные задачи. Не у всех хватало терпения вести утомительную борьбу с постоянными ошибками ввода/вывода.

Несмотря на регулярный технический контроль за машинами, в среднем 2—3 из них постоянно находились в неисправном состоянии. На многие мелочи (не работает клавиша, нет одного из цветов) приходилось просто закрывать глаза.

Об операционных системах «Агата». Существуют две основные ОС для «Агата». Условно будем называть их ДОС «Бейсик» и ДОС «Школьница». ДОС «Бейсик»<sup>1</sup> не только громоздка и неудобна в обращении

(я бы сказал, «недружелюбна»), но и имеет довольно опасные ошибки. Бейсик здесь значительно урезан, но и в таком варианте не обошлось без огрех.

Преподавание с использованием ненадежной машины и недоделанной ОС — чрезвычайно неблагоприятное занятие. Постоянные сбои и неожиданные ошибки дискредитируют в глазах ребят информатику, компьютеры, а иногда и учителя. После того как доверие ученика к машине утеряно, пропадает и его интерес к ней.

К счастью, существует ДОС «Школьница». Со «Школьницей» удобно работать, ее редактор сделан на уровне мировых стандартов.

Заметим, что у «Агата» (да и у любой аналогичного класса машины) язык программирования и ОС тесно связаны друг с другом. Выбрав Бейсик, вы не можете пользоваться редактором «Школьницы» и наоборот. Это влияет на выбор языка для обучения.

Бейсик или Рапира? За полтора года обучения каждый из школьников осваивал в той или иной степени оба языка — Бейсик и Рапиру. Поэтому у нас была возможность оценить, насколько эти языки соответствуют задачам курса информатики, целям школьной компьютеризации и интересам учащихся.

Для того чтобы дать первичные понятия об алгоритмах и программировании, можно использовать любой язык. Но как представить современное программирование без аппарата процедур (которого нет в Бейсике)? Попробуйте также написать на Бейсике рекурсивную функцию (можно, но это будет программистская экзотика). Программирование на Бейсике, скорее, напоминает программирование в машинных кодах: наиболее часто используются три команды — присваивание, условный и безусловный переходы. Бейсик — Агат, собственно говоря, не является языком высокого уровня, а, скорее, представляет собой дополнение к ассемблеру. Написать на Бейсик — Агат приличную программу просто нельзя: вам придется и вручную распределять память, и вставлять куски на ассемблере.

Язык Рапира (в системе «Школьница»), реализованный на «Агате», также имеет некоторые недостатки. К этим недостаткам следует отнести малый объем памяти<sup>2</sup>, предо-

<sup>1</sup> Имеет несколько версий. Сказанное ниже относится ко всем этим версиям. Интересно, что различные версии не полностью совместимы друг с другом.

<sup>2</sup> Интерпретатор Рапиры занимает около половины памяти машины (около 32 К). Это говорит о маленькой памяти «Агата» и искусственности

ставляемой пользователю, низкое быстродействие<sup>3</sup> и отсутствие средств перехвата ошибок.

Рапира богата возможностями организации ветвления, цикла, разнообразием типов объектов. Аппарат процедур очень мощный. Хорошо организована защита имен. Рапира в отличие от Бейсика может научить современному стилю программирования.

У многих программистов, однако, существует предубеждение к Рапире, вызванное необычностью этого языка. Постоянное желание совершить операцию перехода и пристрастие к латинскому алфавиту — скорее вредные привычки, чем предмет гордости. Еще один аргумент противников Рапиры заключается в том, что ее изучить труднее, чем Бейсик. Но ведь и на пальцах считать легче, чем на ЭВМ.

Хочу ответить еще на одно возражение, которое я много раз слышал и от программистов и от учеников: Бейсик широко распространен, а на Рапире никто не работает. Следует помнить, что целью школьного курса информатики является не обучение конкретному языку программирования, а формирование алгоритмического стиля мышления, знакомство с концепцией языков программирования. И Рапира здесь подходит гораздо больше, чем Бейсик: она логичнее, в ней меньше ритуальных действий, ярко выражена блочность построения программ и т. д. Кстати, Е-язык используется еще меньше, чем Рапира, но недаром именно он описан в пробном учебном пособии. Рапира представляет собой в отличие от Бейсика развитие Е — языка.

Богатые возможности и хорошая структуризация Рапиры позволяют легко создавать разные версии языка, вводить учащихся в программирование постепенно, указывая общие и частые конструкции языков программирования.

Опыт показал, что переход с Бейсика на Рапиру довольно сложен для учащихся. Это практически переход на новый стиль программирования. Обратный переход, по-видимому, не составит труда.

Итак, основной курс программирования, преподаваемый на «Агате», на мой взгляд, должен предполагать постепенное изучение Рапиры со все более широким использованием ее возможностей для создания прикладных программ.

**Об обучающих программах.** Высказыва-

емые ниже методические предложения основаны на убеждении, что ЭВМ является не только новым техническим средством обучения, но и довольно интересным интеллектуальным партнером.

Назовем несколько программ-тренажеров (две из них созданы в нашей школе), которые мы использовали в учебном процессе: программы обучения работе с клавиатурой; программа для изучения служебных слов Бейсика; обучение двум методам сортировки массивов; обучение методу деления отрезка пополам для решения уравнений с одним неизвестным. Ясно, что перечисленные программы составляют лишь малую часть потенциального арсенала программ-тренажеров для преподавания информатики. Опыт показал, что программы-тренажеры очень эффективны, поскольку они совмещают игровой и обучающий материал. Правда, большое значение имеет сценарий и оформление такой программы.

Если тренажеры выполняют функцию проверки знаний, то программы другого типа — электронные учебники — должны методично подавать эти знания. Здесь, по-видимому, не обойтись без выдачи текста на экран. Отличие от обычного учебника должно заключаться не только в легком управлении выбором материала, но также в использовании демонстрационных возможностей машины (цвет, мультипликация, звук) и в простоте изменений в таком учебнике.

Перечисленные идеи были реализованы автором этой статьи с помощью двух программ, предназначенных для создания и использования электронных учебников. Создаваемый учебник представляет собой набор текстовых файлов (уроков) на диске, и это позволяет не загромождать память теми разделами учебника, которые не нужны в данный момент.

Одна из программ реализует собственно процесс обучения при наличии на диске соответствующих файлов учебника. Текст учебника содержит специальные символы, которые позволяют управлять порядком, форматом и цветом материала при выдаче на экран, а также обращаться к другим (демонстрационным или тренажерным) программам.

Другая программа представляет собой редактор экранных текстов, она позволяет создавать и редактировать уроки.

Начата работа по созданию учебника — курса знакомства с «Агатом» и Бейсиком. Важной особенностью этого курса является то, что он содержит достаточное количество заданий, которые можно выполнить непосредственно в процессе работы с учебником.

**Методическая концепция обучения**

создателей Рапиры (для сравнения скажем, что интерпретатор языка APL для IBM PC занимает 114 К).

<sup>3</sup> Рапира работает приблизительно в три раза медленнее Бейсика.

программированию<sup>1</sup> Эффективным средством обучения программированию является решение интересной прикладной задачи. Решение такой задачи не должно быть полностью самостоятельным, напротив, алгоритм и решение должны постоянно обсуждаться с учителем, что позволит показать разнообразие способов решения, дать понятие о критериях оптимальности решения.

Речь идет, конечно, о втором этапе обучения, когда ученик получил основные навыки программирования. По-видимому, этот этап наступает в разное время у разных учеников. Учителю требуется определить, на какую самостоятельную работу готов ученик в данный момент. Задачи должны быть индивидуализированы, должны учитывать интересы, способности, уровень подготовки ученика. Группа сильных учеников может получить задачу для совместного решения.

Сознание полезности решаемой задачи является стимулом для ученика. Наоборот, подозрение в том, что программа (задача) никому не нужна, значительно уменьшает интерес к ней. Важно показать ученику, что его программа полезна. Здесь большую помощь могут оказать все учителя школы.

Сложность задачи, которая может быть решена школьником, значительно варьируется. Полезно спросить ученика, какую задачу он мог бы поставить себе сам. По инициативе ребят в школе № 625 были созданы несколько игровых и тестирующих программ. Особо увлеченные ученики самостоятельно осваивают такие тонкости, как распределение памяти или ассемблер. Естественно, что их работа в этом направлении должна только приветствоваться.

Упомянем некоторые программы, которые были созданы нашими учащимися.

Программы десятиклассников: тренажер для устного счета (С. Чугунов); редактор музыкальных мелодий (И. Борисова); решение произвольных уравнений с одним неизвестным (А. Буркин); построение графиков произвольных элементарных функций от одного переменного (А. Буркин); игровое освоение английской лексики (С. Чугунов); пакет программ для ведения и обработки классного журнала (Л. Эпштейн). Програм-

мы девятиклассников: представление целых чисел в произвольной системе счисления (Е. Симонова); программа для проверки усвоения английской лексики (Л. и О. Щелкуновы); перемножение пары многочленов (К. Пирогов); обучение работе с клавиатурой (А. Никитин и М. Тисов); программы для наглядного представления расположения в натуральном ряду чисел, обладающих заданными свойствами.

Самостоятельная работа учащихся может в значительной степени решить проблему создания программного обеспечения школьной информатики. Перечисленные программы являются полезными для школы прикладными программами. Конечно, большинство из них не смогли бы конкурировать с продукцией профессиональных программистов, однако создание школьной библиотеки программ может стать значительным шагом в компьютеризации.

Кстати, о библиотеке программ. Все знают, как утомительна работа по сборке библиотеки, копированию и поиску программ. Сервисная программа для решения этих задач была бы так полезна для школы («букашная» версия такой программы существует).

Перечисленные программы созданы «сильными» учениками. Но и самые «слабые» могут внести лепту в создание школьной библиотеки программ.

**Общие впечатления о школьной компьютеризации.** Корабль школьной компьютеризации перегружен проблемами. Требуются энергичные усилия, чтобы он не затонул в ближайшее время. В первую очередь необходимо решить три основные проблемы. Первая — техническая, а точнее — экономическая. Это проблема надежной школьной ПЭВМ. Тут бессильны Минпрос и АПН, хотя они могут и должны способствовать решению этого вопроса. Вторая — проблема школьного программного обеспечения. Развернуть его разработку можно, создав центры по рекламе и распространению обеспечения, различными способами стимулируя всех его разработчиков. Третья, впрочем, общая для школьной реформы, — повышение инициативы школьных учителей и администраторов. Школьная компьютеризация открывает огромный простор педагогическому творчеству. Но этот простор, как видно, можно и не осваивать.

Эти проблемы требуют широкого обсуждения, тем более что рецептов решения накопилось много. Но главное, они требуют более конкретных крупномасштабных решений.

<sup>1</sup> Излагаемая ниже концепция не является оригинальной Эсм.: Еришов А. П. Г. А. Звенигородский // Информатика и образование. 1987. № 3. С. 6—7). Тем не менее я рискнул кратко изложить ее здесь, поскольку она независимый результат опыта преподавания.

## Информатика на уроках математики

Введение в школе предмета «Основы информатики и вычислительной техники» отразится на содержании и методике преподавания других школьных дисциплин. Если математика развивает у учащихся логическое мышление, то одна из задач информатики — это мышление упорядочить.

Все школьные предметы снабжают учащегося определенной информацией. Ее поток растет с каждым годом и становится в силу различных причин более размытым, без четко определенных для школьников основных направлений.

Как показывает опыт, школьники не всегда умеют применить на практике имеющуюся у них информацию и порой не знают, что они ею располагают.

Выявить в потоке информации главное учащимся поможет информатика — «отрасль науки, изучающая структуру и общие свойства научной информации, а также вопросы, связанные с ее хранением и использованием в различных сферах деятельности» (Советский энциклопедический словарь. М., 1986).

Ниже мы приведем пример того, как информатика может быть использована на уроках математики.

При изучении темы «Алгоритмы работы с величинами» в учебном пособии по информатике А. П. Ершова, В. М. Монахова (ч. 1) подробно рассмотрен пример решения квадратного уравнения. Изучающие эту тему девятиклассники в основном справляются с решением квадратных уравнений, но и у них еще встречаются ошибки.

При повторении темы «Квадратное уравнение» на уроке математики надо еще раз обратить внимание школьников на алгоритм, разобранный в примере 7.1 (учебное пособие, ч. 1, с. 32). Решение квадратных уравнений в IX классе должно быть доведено у учащихся до автоматизма, причем (опять же автоматически) они должны действовать по следующей схеме:

1. Определить числовые значения коэффициентов квадратного трехчлена (это часто вызывает основные ошибки).

2. Вычислить дискриминант.

3. Записать ответ, если надо, произведя вычисления.

Здесь школьник должен действовать, как ЭВМ. Разумеется, многие операции можно производить в уме, но порядок этих опера-

ций может быть только вышеприведенным.

Стоит обратить внимание учащихся на то, что форма ответа нам ясна заранее. Это либо два действительных числа (они могут оказаться равными), либо «решений нет».

При составлении алгоритма требуется знать форму возможных ответов. На уроках математики необходимо научить школьников, еще не решая той или иной задачи, уметь прикинуть, в какой форме может быть получен ответ, а в какой — не может. Так, при решении строгого квадратного неравенства может быть получено либо  $(X_1; X_2)$ , либо  $(-\infty; X_1) \cup (X_2; +\infty)$  (возможно,  $X_1 = X_2$ ), либо  $R$ , либо пустое множество, и ничего больше.

Очень важно научить школьников пользоваться вспомогательными алгоритмами. Это дало бы возможность быстро рассматривать на уроке математики примеры, не вдаваясь в частные вычисления. При изучении темы «Квадратные неравенства» можно использовать вспомогательный алгоритм решения квадратного уравнения. Тогда, чтобы решить квадратное неравенство, можно:

1. Исполнить вспомогательный алгоритм решения квадратного уравнения.

2. Сделав соответствующий чертеж, записать ответы. Например:

а)  $3x^2 + 5x - 8 > 0$ .

Первый этап: решение квадратного уравнения

$$3x^2 + 5x - 8 = 0.$$

$$x_1 = 1; \quad x_2 = -\frac{8}{3}.$$

Второй этап: чертеж параболы, ветви вверх. Две точки пересечения с осью (OX)

Ответ:  $(-\infty; -\frac{8}{3}); (1; +\infty)$ .

б)  $-x^2 + x + 2 > 0$ .

$$x_1 = -1; \quad x_2 = 2.$$

Ветви вниз. Две точки пересечения.

Ответ:  $(-1; 2)$ .

в)  $5x^2 - x + 11 > 0$ .

1. Уравнение решений не имеет.

2. Ветви вверх, точек пересечения нет. Вся парабола над осью OX.

Ответ:  $R$ .

г)  $-3x^2 - x - 1 > 0$ .

1. Уравнение решений не имеет.

2. Ветви вниз, точек пересечения нет, вся парабола под осью OX.

Ответ: пустое множество.

На уроке нужно объяснять, что метод подстановки при решении любых уравнений и неравенств есть не что иное, как воз-

возможность использования вспомогательного алгоритма.

Например:  $x^4 - 5x^2 + 4 = 0$ .

Вспомогательный алгоритм решения квадратного уравнения

$$t^2 - 5t + 4 = 0.$$

Если есть решения, то вспомогательный алгоритм решения неполного квадратного уравнения  $t^2 = a$  (в данном случае  $t^2 = 4$ ;  $t^2 = 1$ ). Иначе — решений нет.

$$4x^2 - 3 \cdot 2x + 2 = 0.$$

1. Вспомогательный алгоритм решения квадратного уравнения

$$t^2 - 3t + 2 = 0.$$

Если решения есть, то вспомогательный алгоритм решения простейших показательных уравнений  $2^x = a$  (в данном случае  $2^x = 1$ ;  $2^x = 2$ ). Иначе — решений нет.

Решение простейших показательных уравнений тоже может быть представлено в алгоритмической форме:

$$a^x = b \quad (a > 0; a \neq 1).$$

**Если**  $b > 0$ , то  $x = \log_a b$ ;

**иначе** — «решений нет».

Аналогично обстоит дело с решением простейших тригонометрических уравнений. Например:

$$\cos x = a.$$

**Если**  $|a| \leq 1$ , то  $x = \pm \arccos a + 2\pi n \quad n \in \mathbb{Z}$ .

**Иначе** «решений нет».

Можно создать библиотеку алгоритмов по математике, аналогичную той, которая имеется в конце учебного пособия по информатике. Это поможет школьникам не только упорядочить свои знания, определить основные знания и умения, но и уяснить общие закономерности вычислений человека и ЭВМ.

Такие алгоритмы могут быть составлены для курсов алгебры, начал анализа, геомет-

рии. Например, алгоритм решения прямоугольных треугольников, алгоритм построения (нахождения) линейного угла данного двугранного угла, алгоритм построения (нахождения) угла между наклонной и плоскостью и т. п.

Если учащиеся, рассматривая задачу «Диagonalь правильной четырехугольной призмы наклонена к основанию под углом  $\alpha$ . Под каким углом наклонена к основанию плоскость, проходящая через противоположные ребра нижнего и верхнего оснований?», сообразят, какие алгоритмы им понадобятся, и применят их, задача будет успешно решена. Кстати, это может упростить и запись решения.

Разумеется, алгоритмы не должны даваться школьникам как нечто готовое. Учащиеся сами в процессе обучения составляют тот или иной алгоритм, а потом доводят процесс его исполнения до автоматизма.

В методике преподавания математики алгоритмам уделялось то большее, то меньшее место. Надо полагать, что в информатике им будет уделено достаточное внимание.

За последние двадцать лет в школьном курсе математики все реже встречались задачи с параметрами. Изучение команды ветвления затруднительно без опыта решения такого рода задач девятиклассниками. Возможно, на уроках математики в младших классах стоило бы познакомить школьников с решением задач с параметрами. Далеко не всякий школьник сможет, например, исследовать в общем виде линейное уравнение и тем более линейное неравенство. Все это вызывает трудности на уроках информатики, например, при изучении алгоритма решения линейного уравнения  $ax = b$ , где  $a$  и  $b$  — произвольные вещественные числа.

## Вниманию читателей

Казанский городской компьютерный клуб совместно с Центром информатики и вычислительной техники и комитетом ВЛКСМ Казанского государственного университета им. В. И. Ульянова-Ленина с 7 по 10 июля 1988 года проводят в Казани

Всесоюзную встречу-слет представителей компьютерных клубов.

Для участия в слете необходимо выслать заявку по адресу:

422520, Татарская АССР, г. Зеленодольск, а/я 161.

Оргкомитет слета.

Справки по тел. 32-39-34. Ответственный Сергей Фомин.



Журнал в журнале  
для школьников, студентов,  
учащихся СПТУ и техникумов

Издается при участии ЦК ВЛКСМ

# МОЛОДЕЖНАЯ ИНИЦИАТИВА

## 3

### В НОМЕРЕ:

- ПРИГЛАШЕНИЕ К УЧАСТИЮ В КОНКУРСЕ
- ДЕТСКИЙ КЛУБ «РАДУГА»
- СОВЕТСКИЙ ШАХМАТНЫЙ КОМПЬЮТЕР
- КАК В СПТУ-25 ПОСЕЛКА ВАНГАЖИ РАБОТАЮТ С КОМПЬЮТЕРАМИ
- ЭВМ В ШВЕЙНОМ ПРОИЗВОДСТВЕ
- КЕМ БЫТЬ! ПРОГРАММИСТОМ!

Шесть лет... Восемнадцать... Тридцать... Проходят годы, взрослеет человек, становится мудрее, опытнее...

А чем измерить возраст издания?

Номерами!

Вы открыли сейчас шестой выпуск «Молодежной инициативы». Публикации для молодежи, они и подготовлены молодыми, теми, кто сегодня открывает для себя увлекательный, полный загадок и открытий мир информатики.

«Молодежная инициатива», становясь старше и опытнее, во многом поможет своим читателям: это — ее задача. Но и ей нужна Ваша помощь!

Присылайте свои материалы, фотографии, рисунки, идеи.

Ждем ваших писем!

## Всесоюзный конкурс «Юный программист»



демонстрационные, обучающие, контролируемые программы по предметам, исполнители, игровые программы и т. д.

При разработке программ следует ориентироваться на школьные персональные ЭВМ: «Электроника БК-0010» в комплекте с ДВК-2М (КУВТ-86), «Корвет», УКНЦ, «Ямаха», «Агат».

Учреждено 6 главных призов и 10 поощрительных.

Для участия в конкурсе до 1 февраля 1989 г. необходимо прислать заявку и подробный сценарий программы. Сценарий должен содержать основные идеи и подробное описание программы.

Авторы отобранных по итогам конкурса сценариев будут приглашены в Красно-

ярск в конце марта 1989 г. для участия в заключительном туре конкурса.

Материалы на конкурс направлять по адресу: 660051, г. Красноярск, абонемента № 11533. Оргкомитет конкурса «Юный программист».

### ЗАЯВКА

Ф. И. О. \_\_\_\_\_  
Домашний адрес \_\_\_\_\_  
Номер и адрес школы (СПТУ) \_\_\_\_\_

Класс \_\_\_\_\_  
Тип машины \_\_\_\_\_  
Язык программирования \_\_\_\_\_

Название программы (кому и для чего предназначена) \_\_\_\_\_

Объем программы \_\_\_\_\_  
Среднее время работы программы \_\_\_\_\_

104

Красноярский крайком ВЛКСМ, Вычислительный центр СО АН СССР (г. Красноярск), Красноярский государственный педагогический институт объявляют Всесоюзный конкурс «Юный программист» на лучшую программу, созданную школьниками для детей до 12 лет.

На конкурс принимаются клавиатурные тренажеры,

## «Радуга» в Мытищах

Детский клуб «Радуга» подобен многим, имеющимся чуть ли ни при каждом жилищно-коммунальном хозяйстве. Но от остальных его отличает то, что здесь наряду с кружками фотодела, рисования и спортивными секциями есть компьютерный класс. Ребята изучают здесь информатику.

И потому первым делом я спросила одного из членов клуба, ученика IX класса Славу Шендрикова, чем занятия в компьютерном классе «Радуги» отличаются от школьных уроков информатики.

«Ну во-первых,— сказал Слава,— в школе все это намного скучнее: изучаем в основном одну теорию. Здесь же мы большую часть заня-

тый работаем с компьютером, сами разрабатываем программы, играем. Разговаривай сколько хочешь, советуясь с товарищами...»

Можно еще добавить, что в компьютерный клуб принимают ребят с V класса. Многие из пятиклассников уже могут самостоятельно составлять программы для ЭВМ. Всего же здесь занимается около тридцати школьников. Занятия проходят два раза в неделю плюс еще день, когда можно просто прийти поиграть на компьютере. Занятия проводят опытные специалисты по вычислительной технике, двое из них — кандидаты технических наук. Их энтузиазму и обязан клуб своим рождением. Ими же разра-

ботаны программы обучения в классе. Но сами преподаватели считают что главное не программа, а методика обучения. Важно заинтересовать ребят. Уроки в классе строятся так: деловая часть чередуется с игровой. Я спросила у председателя клуба кандидата технических наук В. Коллегова, в чем видят организаторы свою задачу.

«Прежде всего,— сказал Владимир Григорьевич,— через компьютерное обучение поднять уровень технической грамотности ребят, уровень культуры мышления. Сегодня практически все ребята, занимающиеся в нашем клубе, освоили основы программирования, изучили особенности языка Бейсик, получили знания и практические

навыки, закрепили их в процессе написания программ. Машины «Агат», которых в нашем клубе установлено шесть штук, в методическом плане очень удобны. Но, пожалуй, самое главное, чему мы стараемся научить ребят, это умению правильно ставить перед собой задачу. Ведь в жизни не всегда окажется под рукой учебник, где говорится, что то-то и то-то нужно сделать так-то. Любой специалист, по моему мнению, должен уметь прежде всего точно определить цель, найти пути ее решения».

Разработали члены клуба программы для ЭВМ. И возникла перед ними новая задача — найти потребителя для программ, чтобы ребята смогли почувствовать значение своего труда. Сегодня в клубе прочно утвердилось два принципа: самоуправления, помогла бы окупить все эти расходы да еще заработать дополнительные средства на развитие клуба.



ни одно начинание, даже организация в детском клубе компьютерного класса, не может обойтись. Ведь, скажем, ДЭЗ № 3 Главмостроя, финансировавшей организацию клуба, создание компьютерного класса обошлось в несколько тысяч рублей. Возможность продажи программ, созданных ребятами, помогла бы окупить все эти расходы да еще заработать дополнительные средства на развитие клуба.

А учитывая то, что многие отечественные ЭВМ еще не всегда достаточно обеспечены пакетами программ, выгода могла бы быть обоюдной. Решением этих проблем и занимаются сегодня руководители клуба.

Многие девятиклассники, занимающиеся в компьютерном классе клуба «Радуга», мечтают связать свою будущую специальность с информатикой.

**Т. КРАСНОВА**

## До встречи в Нью-Васюках!

Успехи советских «электронных шахматистов» общеизвестны. Именно советская шахматная программа «Каисса» стала первым чемпионом мира среди ЭВМ.

Скоро на прилавках магазинов появятся ее младшие братья — специализированные шахматные компьютеры. К производству готовы две модели — «Стратег» и «Электроника-01».

«Электроника-01» содержит однокристалльный микропроцессор, ОЗУ на 2К байт и ПЗУ на 16К байт. Она может начинать партию сначала, продолжать с произвольной позиции, способна играть сама с собой. Шесть уровней игры — в зависимости от времени обдумывания хода (от 10 с до 20 ч) — позво-

ляют ей подстраиваться под силу игры партнеров, вплоть до второго разряда. Ходы в виде шахматной нотации вводятся с помощью 24 клавиш, ответные ходы высвечиваются на 5-разрядном индикаторе.

Как видите, игрушка неплохая, правда, не без недостатков. Первый — размеры, далеко не карманные: вес ее 2,5 кг. Второй — далекая от «карманной» цена: предположительно 200—300 руб.

Как кому, а школьникам пока предпочтительней живые партнеры!

Будем надеяться, что в скором времени центры инерформатики закупят такие компьютеры, цены на пользование будут, наверно, до-

ступными, и наиболее инициативные из наших читателей напишут нам о своих победах над ЭВМ.

**К. ШЕХОВЦЕВ**



## Что может «Лектра»?

Лазерные установки, машины с числовым программным управлением... Тихо мерцают экраны дисплеев... Цех конструирования мужских костюмов Ивановской фабрики им. Ленинского комсомола напоминает скорее научно - исследовательскую лабораторию. А стал он таким совсем недавно. Ивановские швейники заключили договор с французской фирмой «Вестра Юнион» на изготовление мужских костюмов по лицензии этой фирмы.

Все оборудование для новой линии поставлено французскими коллегами, в частности система «Лектра», предназначена для раскладки, вырезки и размножения лекал. Ее можно назвать электронным мозгом новой линии, ибо именно конструирование будущих костюмов является основой всего процесса.

Нажатие кнопки, и на экране дисплея одна за другой появляются детали будущего

костюма. С помощью клавиатуры их можно видоизменять, доводить до кондиции. В памяти компьютера — раскладки всех размеров с 44-го по 68-й. За несколько секунд можно воспроизвести любую из них. Раньше их изготавливали вручную по картонным образцам, которые за время длительного хранения могли усохнуть, покоробиться. Теперь же при помощи компьютера ELA-73 соответствие деталей образцу можно установить с точностью до микрона.

После того как раскладка на компьютере закончена, на лазерной установке, работающей в автоматическом режиме, изготавливаются лекала. По заданной программе невидимый луч с точностью до микрона вырежет любую деталь.

«Оператор по изготовлению лекал» — так теперь называется профессия работников в цехе конструирования. Обучение работе на

системе «Лектра», которое проводили французские специалисты, продолжалось три месяца. При отборе на эту специальность учитывались два условия: образование не менее десяти классов и опыт работы по изготовлению лекал. Ну и конечно же, желание.

Сегодня все работницы полностью овладели технологией, а если случаются какие-то затруднения, то достаточно заглянуть в конспекты или проконсультироваться у конструктора по изготовлению мужских костюмов Ю. Ваулина, который обучался работе на «Лектре» непосредственно во Франции.

Высокое качество продукции — вот что дает ивановским швейникам система «Лектра» и новое, современное оборудование, пришедшее в цеха фабрики с пуском советско-французской линии.

Т. БЕЛОВА

106

## ЭВМ на уроке и после него

В СПТУ-25 поселка Вангажи, что находится недалеко от Риги, один из лучших в республике кабинетов информатики. Молодой преподаватель Петерис Цайцс (ему всего 25 лет, он выпускник ЛГУ им. П. Стучки) приложил немало усилий, чтобы в училище, где готовят электромонтеров, газоэлектросварщиков, слесарей по ремонту оборудования, информатика изучалась по-настоящему. Он легко находит общий язык с ребятами, поскольку их связывают не только обучение, но и увлечения: Пе-

терис ведет музыкальный кружок и дискотеку, где с выдумкой применяется электронно - вычислительная техника. Например, с учащимися сделали установку «Бегущие огни». Вместе со старшим братом Янисом, который работает здесь же заместителем директора по УПР и преподает физику, Петерис играет в инструментальном ансамбле.

Руководит училищем А. И. Шафранович, который умеет и любит работать с молодежью. Он давно понял необходимость внедрения

ЭВТ в учебный процесс училища. В каком еще СПТУ есть свой пусть маленький, вычислительный центр, где создаются программы, обеспечивающие контроль за динамикой учебного процесса?! Модно, престижно? Да нет, просто нужно!

А какие проблемы? Да как у всех... Например, не хватает дисков для ДВК-2М.

Тем не менее создаются интересные программы, которые обеспечивают межпредметные связи. Так, на электротехнике (ее преподает К. Харьо) применяется



В СПТУ-25 творчески подходят к использованию вычислительной техники, как на занятиях, так и во внеклассной работе



«Бегущие огни» — программа, созданная для дискотеки

программное управление тренажером «Ошибки при сборке схемы реверса асинхронного двигателя» (разработчик П. Цайцс).

Где любят свое дело, вкладывают, не считаясь с личным временем, силы, энергию, знания в работу, там, как видим и результаты.

Вот он положительный опыт — приезжайте и изучайте, здесь вам будут рады.

Т. КИРИЧЕНКО

107

## Информация к размышлению

Кем быть? Этот вопрос современный молодой человек должен решать рационально, взвесив все обстоятельства, в том числе и такое — насколько велика потребность общества в специалистах того или иного профиля.

В Москве недавно выпущен уникальный справочник — «Бюллетень вакантных рабочих мест для специалистов и служащих». Он интересен по двум причинам. Первая — создали его при помощи ЭВМ в Вычислительном центре Мосбыта. Представляет, как облегчается участь человека, ищущего работу, если под рукой оказывается такой справочник, кстати, доказывающий, что безработицы у нас нет! В этом издании указан и район, где требуются специалисты и оклад вакантной должности.

Вторая причина, побудившая нас опубликовать информацию об этом справочнике чисто профориентационная.

Сообщаем, что больше всего не хватает программистов, операторов ЭВМ, секретарей-машинисток и бухгалтеров, а также инженеров по электронике, учителей и экономистов. Словом, выбирая профессию, думай не только о себе.

Купить новое издание пока нельзя, но в скором времени, возможно, его будут распространять через киоски «Союзпечати».

А. РОДИОНОВ,  
Н. РОДИОНОВА

## Диалог в учебных программах

108

Один из самых больных, по нашему мнению, вопросов в разработке педагогических программных средств (ППС) для персональных компьютеров — организация диалога. Анализ отечественного программного обеспечения, создаваемого для школ, вузов и СПТУ, показывает, что существуют как минимум две опасные тенденции, влекущие за собой ослабление интереса педагогов и учащихся к ППС и возрастание сомнений в необходимости использования компьютеров в учебном процессе.

1. Учебная программа превращается в «электронную книгу», чтение которой отличается от чтения обычной книги только тем, что вместо привычного перелистывания страниц учащийся нажимает соответствующие кнопки на клавиатуре компьютера. При этом иллюстративный графический материал появляется на экране с удручающе низкой скоростью, так как обычно он порождается средствами интерпретатора с языка Бейсик.

2. Учебная программа насыщается чрезмерно большим количеством вопросов, ответы на которые учащийся должен вводить с клавиатуры компьютера, используя некоторое подмножество слов русского языка. Программы при этом не осуществляют достаточный лексический контроль вводимого текста, тем более что такой контроль практически невозможен из-за орфографических и других ошибок, допускаемых учащимися (в каждое предметное ППС пришлось бы заложить курс русского языка). Поскольку не все учащиеся достаточно свободно владеют алфавитно-цифровой клавиатурой компьютера, работа с такими ППС превращается для них в процесс поиска букв на клавиатуре, а в дальнейшем в не очень интересную игру в угадывание слов.

Как же найти выход из этой ситуации? Прежде всего следует обратить внимание разработчиков на то, что с подобными проблемами столкнулась первая отнюдь не школьная информатика. Ранее этот вопрос был поставлен и в целом успешно решен в конторских системах и системах деловой графики, а также в чисто игровых программах. И в том и в другом случае появление персональных компьютеров, обладающих графическими средствами отображения информации на экране монитора, изменяло человеко-машинный интерфейс. Но в отличие от учебных программ, задача поддержания максимально эффективного диалога в интерактивном режиме здесь стояла более остро — деловые люди не любят терять время на болтовню (пусть даже с компьютером), а играющие в компьютерные игры предпочитают такие, которые обладают быстрой и динамичной графикой и используют ограниченное количество органов управления игрой. Как следствие, в деловых сферах применения компьютеров появились различные многооконные системы с быстрым и в значительной мере фиксированным выбором предлагаемых альтернатив и графически обозримой иерархически вложенной системой работы со сложными структурами данных и документами (концепция «рабочего стола», с перекладываемыми на нем папками и листами бумаги), а в играх — строго фиксированный выбор из небольшого числа альтернатив (вправо — влево, вверх — вниз, стрелять — не стрелять, прыгать — не прыгать и т. п.) в быстро меняющейся ситуации.

Характерным для современного состояния и того и другого класса программ является взаимное проникновение методов ведения диалога из одного класса в другой: в сложных играх все чаще начинают использовать мно-

гооконный диалог, а в деловых системах стремятся свести к минимуму число органов управления за счет повышения «интеллектуальности» программ. Это говорит о том, что в первую очередь разработка программ ведется с учетом удобства работы с ними пользователя. Если же об этом забывают, то появляются программы на вполне современных компьютерах (например, на «Ямахе»), которые по реализации находятся на уровне техники программирования 60-х годов. И эти программы предназначены для учащихся! При этом возникает законный вопрос: нужен ли компьютер в школе, если с ним неудобно работать? Но неудобно работать не с компьютером, а с ППС, которые разрабатывались не для человека, а для компьютера!

Другой вопрос, который может задать лю-

бой разработчик ППС: где же системные программисты, которые должны создать необходимые нам инструментальные средства (конечно же не в рамках языка Бейсик) и обучить нас пользоваться ими?

И третий вопрос задают авторы этой статьи, один из которых является системным программистом, а другой — разработчиком ППС: где же авторское право на такие программные продукты и их соответствующая оплата? Ведь системные программы создаются не обязательно в рабочее время, а с появлением персональных компьютеров — и не обязательно на казенной технике!

Работа с хорошими обучающими программами агитировала бы за использование компьютеров в школе гораздо лучше, чем множество слов, уже сказанных на эту тему!

**Л. ВИКЕНТЬЕВ,**  
доктор техн. наук, профессор  
**О. КОЗЛОВ,**  
канд. техн. наук

## Наше мнение

Отрадно, что редакция журнала предоставляет возможность высказать различные взгляды на обучение основам информатики: этому теперь отведена целая рубрика. Нам хотелось бы обсудить опубликованные в «Точке зрения» материалы А. Михайлова (1987, № 5) и А. Матюшкина-Герке (1987, № 6).

Мы преподаватели основ программирования и применения ЭВМ в техническом вузе, кроме того, нам часто приходится принимать участие в различного рода конференциях и семинарах по вопросам как вузовской, так и школьной информатики. К сожалению, в статьях А. Михайлова и А. Матюшкина-Герке мы услышали «знакомые мотивы». А. Михайлов, наверное, является программистом весьма высокого уровня, что и накладывает отпечаток на его взгляды, но, как отмечено в других статьях этой рубрики, в школе следует готовить не специалистов по программированию, а людей, умеющих применять вычислительную технику в своей повседневной профессиональной деятельности. Поэтому многие высказанные им положения мы считаем в корне неверными, о чем и хотим предупредить учителей информатики, не искушенных еще в тонкостях этого предмета.

Основное возражение вызывает мысль А. Михайлова о том, что наивной является уверенность в возможности обучения работе с компьютером каждого. Мы со всей ответственностью заявляем, что работе на машине действительно может научиться любой нормальный человек, знающий буквы, цифры и четыре действия арифметики. За многие годы работы мы не встречали школьников, студентов или специалистов, которые, желая освоить основы компьютерной грамотности, не смогли бы этого сделать. Практическая работа на ЭВМ не искусство, а ремесло, научиться которому может каждый. Достичь же вершин мастерства в своем деле действительно дано не всякому, и программирование в этом смысле не исключение.

Путь, предложенный А. П. Ершовым, — через алгоритмизацию к алгоритмическому мышлению — совпадает с нашими взглядами на основное содержание школьного курса. Поэтому изучение основ алгоритмизации с помощью русского алгоритмического языка, особенно усиленное программной поддержкой, отнюдь не проявление ложного патриотизма, а объективная необходимость.

Мы убеждены, что в нашей стране основам компьютерной грамотности необходимо обучать только на русском языке — языке

межнационального общения народов СССР, чтобы за «деревьями» разноязычных слов обучаемые не потеряли общей картины «алгоритмического леса».

Имеются ли у русского алгоритмического языка, излагаемого в пробном учебном пособии, недостатки? Вопрос этот является, безусловно, риторическим. К недостаткам этого языка относится в первую очередь, как это отмечается в статье А. Матюшкина-Герке, отсутствие средств «для описания взаимодействия между пользователем и исполнителем». Однако выход, по нашему мнению, не в том, чтобы отменить изучение русского алгоритмического языка, а в том, чтобы пополнить изучаемый язык операторами обмена информацией, например операторами ввода — вывода. Кроме того, мы считаем необходимым дополнить язык такой конструкцией, как комментарий, в ряде случаев было бы полезно использовать и оператор перехода (например, для организации выхода из цикла при различных условиях).

И мы уверены, что если в учебной программе, предложенной А. Матюшкиным-Герке, при изучении алгоритмизации сначала изложить расширенную версию русского алгоритмического языка, а в последующем показать, как должен осуществляться переход от конструкций такого языка к программам на производственных языках программирования (в рассматриваемом варианте это язык ПМК и Бейсик), то она станет гораздо легче для усвоения, чем исходная. Это объясняется тем, что в обсуждаемой учебной программе Бейсик и язык программирования для ПМК рассматриваются как

два независимых, практически несвязанных объекта, а в «модифицированном» варианте эти языки должны рассматриваться только в качестве средства для реализации конструкций русского алгоритмического языка.

Исходя из того, что обучать компьютерной грамотности необходимо всех, а не только владеющих английским языком, мы считаем, что наибольшую помощь массовому пользователю профессиональные программисты могли бы принести разработкой для каждой пары *процессор — операционная система* соответствующего перекодировщика, обеспечивающего перевод сообщений системы с английского языка на русский.

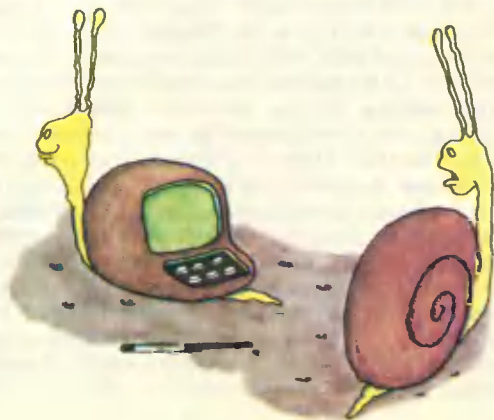
Хотелось бы затронуть еще один, к сожалению, все еще очень актуальный аспект, связанный со статьей А. Михайлова. Сам по себе тезис о том, что безмашинное обучение основам компьютерной грамотности хуже машинного, напоминает утверждение, что лучше быть здоровым и богатым, чем бедным и больным. Сегодня речь может идти только о том, что хуже: безмашинное обучение или полное отсутствие обучения. Выбирая из двух зол меньшее, мы и получаем сегодняшний вариант безмашинного обучения. Но уже сегодня альтернативой ему может стать широкое использование программной поддержки русского алгоритмического языка на ЕС ЭВМ и выполнение ряда учебных задач в пакетном режиме на ближайших недогруженных вычислительных центрах. Такой подход менее эффективен, чем работа на ПЭВМ, но и он может решить важную задачу: показать школьникам, что машина действительно может стать их надежным помощником.



### СОВЕТЫ, СОВЕТЫ, СОВЕТЫ...

- Не суй нос не в свою программу.
- Не допускай вольных переводов твоих программ транслятором.
- Не учи программированию своего начальника.
- Не загрязняй распечатками окружающую среду.
- Не допускай, чтобы один вид многомашинных комплексов вызывал в тебе комплекс неполноценности.

- Не называй программы именами своих руководителей.
- Не забывай, что неотлаженная в срок программа требует срочного налаживания отношений с начальством.





# ЭВМ в народном хозяйстве

**В. ФОМИН,**

зав. отделом Центрального музея Революции СССР

## Компьютеры в музее

В нашей стране научно-исследовательские работы, связанные с применением компьютеров в музеях, начались в конце 70-х гг., когда в Западной Европе, США и Канаде крупные музеи уже имели собственные ЭВМ или были подключены к компьютерным сетям. Например, во Франции и Канаде существуют развитые компьютерные сети, объединяющие музеи по всей территории этих стран; применение ЭВМ позволяет оперативно получать информацию о любом музейном экспонате. Очевидна важность такой информации для создания новых экспозиций, выставок, изучения и пропаганды исторического наследия.

Использование вычислительной техники особенно актуально для нашей страны, располагающей музейным фондом в 55 млн. предметов. Посещаемость советских музеев в 1986 г. составила огромную цифру — 182 млн. человек. «Музейный бум», несомненно, способствует тому, что большинство крупных музеев страны работает на пределе своей пропускной способности. Кроме того, издания о памятниках истории и культуры, каталоги и альбомы музейных коллекций пользуются огромным спросом и популярностью. Но значительная часть музейного фонда по различным причинам пока остается труднодоступной для посетителей и специалистов, обращающихся к музейным коллекциям по профилю своей работы.

Решить эти проблемы поможет вычислительная техника, которая значительно расширит доступ к фондам музеев. Ведь в музейных экспозициях обычно представлено только 10—12 % общего числа предметов, хранящихся в фондах. А в крупнейших музеях СССР (Государственный Эрмитаж, Государственная Третьяковская галерея, Госу-

дарственный Исторический музей, Центральный музей Революции СССР) экспонируется всего 3—5 % музейного собрания.

Для музейной работы важно иметь необходимый и одновременно достаточный объем информации о всех музейных предметах в каждом музее страны. Эту информацию будет обрабатывать вычислительная техника, с помощью которой можно создавать банки данных о коллекциях. К их разработке приступил Центральный музей Революции СССР — крупнейшее хранилище памятников истории и культуры, рассказывающих о революционной борьбе до Великой Октябрьской социалистической революции, строительстве социалистического общества в нашей стране.

Почти миллионный состав коллекций, научно-исследовательская работа сотрудников и публикация ее результатов в собственных изданиях музея, большая выставочная деятельность (ежегодно музей организует несколько десятков зарубежных и внутрисоюзных выставок), ежегодное обслуживание более 1,5 млн. посетителей, создание новых разделов экспозиции, десятки ежедневных запросов от советских и зарубежных организаций и частных лиц требуют качественных изменений в поиске и обработке музейной информации. И здесь компьютер незаменим. Поэтому еще в 1980 г. Министерство культуры СССР предложило Государственному Эрмитажу и Центральному музею Революции СССР исследовать возможности применения ЭВМ в музейном деле и создать комплекс действующих моделей для отработки всей технологической цепочки подготовки, ввода и получения информации. В разработке математического и системного обеспечения музеев помогал

институт «Гипротейтр». Кроме того в музеях были созданы специальные подразделения музейной информатики. Они проводили анализ научно-исследовательской, методической, фондовой, экспозиционной, выставочной, пропагандистской деятельности Центрального музея Революции СССР. Определены ее перспективы до 2000 г. с учетом развития музейного дела в СССР и за рубежом.

В результате выделены первоочередные задачи, для решения которых необходимо использовать вычислительную технику. Это учет поступлений и выдач (постоянных и временных), движения музейных фондов по всему собранию, отдельным фондам и коллекциям, статистическая отчетность для Министерства культуры СССР и Государственного комитета по статистике СССР, научная обработка музейных предметов и их изучение, получение информации о музейных фондах для подготовки материалов, издания каталогов, указателей, обзоров, путеводителей, монографий, статей. Кроме того, это информационное обслуживание ученых и потребителей в музее и вне его, планирование и управление основными направлениями деятельности музея (научно-исследовательской, методической, экспозиционной, выставочной, пропагандистской, фондовой, реставрационной, издательской и т. д.), а также управление кадрами, бухгалтерией, финансами, снабжением музея.

Решение приоритетных задач поможет повысить оперативность получения, достоверность и точность информации о музейных коллекциях, создать предпосылки для повышения эффективности и своевременности принятия решений об учете музейных коллекций. Это улучшит контроль за сохранностью и комплектованием всего музейного собрания, снизит трудоемкость и ускорит подготовку выпуска каталогов и указателей. Музейные специалисты будут освобождены от выполнения рутинных операций (многократной переписки данных, ручного поиска информации в картотеках, составления различных списков, сводок и т. д.), что позволит им сконцентрировать внимание на решении научных и творческих проблем.

Компьютеры могут применяться и для унификации учетной документации. Для этого в музее введен единый учетно-хранительский документ, имеющий 52 признака описания музейного предмета. Такая форма документа позволила разработать четкие методики описания музейных предметов коллекции. Опыт внедрения нового учебно-хранительского документа был использован и для создания унифицированного паспорта музейного предмета, введенного в 1987 г.

во всех музеях страны.

Еще одна задача, которую можно решать с помощью ЭВМ,— стандартизация терминологии научного описания музейных предметов. Надо создать систему словарей музейных терминов по всем коллекциям, информация о которых вводится в память компьютера. Такие словари будут созданы для крупнейших наших музеев, например Центрального музея Революции СССР, Государственного Эрмитажа, Государственной Третьяковской галереи, Государственного Исторического музея, Государственного Русского музея, Музея этнографии народов СССР, Государственного музея декоративно-прикладного и народного искусства РСФСР. Каждому из них поставлена конкретная задача по разработке определенной части музейной терминологии, чтобы избежать ненужного дублирования.

Центральный музей Революции СССР, создавая автоматизированную базу данных о коллекциях основного фонда, разрабатывает наборы терминов, которые используются при вводе информации в память ЭВМ.

Вот примеры признаков, где используется стандартная терминология: способ поступления, классификация, типология, народ, этногруппа, хронология, место создания, материал, техника, время и место бытования, время и место обнаружения, исторические события, связанные с памятником, и рекомендации по реставрации, возможность транспортировки. Число терминов, используемых в этих признаках, различно и колеблется от 2 до 100. В ходе исследовательской работы их количество будет возрастать.

Важная проблема, которую приходится решать при создании автоматизированной базы данных для музея,— взаимоотношения человека и компьютера. Обычно научные сотрудники музея с гуманитарным образованием не имеют опыта работы с ЭВМ. Поэтому должны быть созданы программы, предельно упрощающие процесс общения музейного работника с компьютером.

При создании автоматизированной базы данных в нашем музее это обстоятельство учитывалось особо. Разработчикам программного обеспечения удалось создать систему, реализующую диалог человека и ЭВМ, на русском языке.

Несколько слов об ЭВМ для музея. Для успешной работы с информацией о музейных коллекциях требуется компьютер с большим объемом памяти и наличие специальных программ.

Сейчас наш музей использует мини-ЭВМ, «Реалите» ИН-500, которая установлена в Вычислительном центре Министерства культуры СССР. Этот компьютер обладает

развитым сервисным оборудованием, предоставляет возможность работать с ЭВМ в режиме диалога при обработке и корректировке специальной информации о музейных коллекциях.

Удачная разработка математического обеспечения системы, использование унифицированных документов и стандартной терминологии позволили при создании автоматизированной базы данных музея использовать адаптированный русский язык, который удобен для общения с компьютером и не требует от пользователя знания языков программирования. ЭВМ сама подсказывает порядок действий при вводе, коррекции или получении информации. Подсказка может быть в виде сообщения на экране дисплея о порядке нажатия клавиш, в виде текстов словарей при заполнении определенных признаков описания, сообщения о допущенной ошибке или переходе к другому этапу работы с базой данных.

Автоматизированная база данных о коллекциях основного фонда Центрального музея Революции СССР включает, в частности, информацию о следующих коллекциях: отечественные и зарубежные знамена и флаги, фарфор и живопись.

Система позволяет получать промежуточ-

ную и выходную информацию в виде документов на бумажных носителях, сообщений на экране дисплея, документов на машинных носителях для обмена информацией с автоматизированными базами данных других музеев СССР.

Информация по желанию пользователя может быть выдана в виде полного или краткого описания музейного предмета, списка (табличная форма) или справки.

Для ускорения поиска нужной информации в системе используется 18 признаков, по которым возможен быстрый поиск. Они могут быть заданы последовательно или в любом сочетании. Такая гибкая система позволяет получать каталоги коллекций и указатели к ним, справки о коллекциях основного фонда, таблицы и т. д.

Хотя процесс формирования автоматизированной базы данных о коллекциях музея еще только начинается, уже сейчас можно говорить о выполнении основных требований, которые ставились перед системой. Это однократность использования входных документов и многократность использования введенной информации, ее актуальность, обеспечение оперативного доступа к ней, сохранность информации и ведение архивов, защита ее от несанкционированного доступа.

## Музей в компьютере

Во Всесоюзном музейном объединении «Государственная Третьяковская галерея» существует отдел информатики. Здесь проходит испытания новейшее электронное оборудование — основа будущего компьютерного центра, который со временем войдет в общесоюзную информационно-поисковую музейную систему памятников образительного искусства.

На первом этапе создания системы в нее вводят сведения о коллекции Третьяковской галереи. Каждый экспонат — а их более 70 тыс. — будет представлен в памяти ЭВМ более чем сотней различных характеристик: описанием, сведениями об авторе, историей создания, экспозиционной судьбой и многими другими. Самое интересное — на экране дисплея при желании можно будет воспроизвести цветное изображение экспоната. Широкий доступ к такому богатому банку данных делает возможным для очень многих людей несравненно более глубокое, чем раньше, общение с

искусством. Самим же музейным работникам будет проще получить срочно понадобившуюся информацию, станет легче планировать выставки, во многих случаях удастся обойтись обращением к информационной системе вместо того, чтобы доставать подлинники из запасников.

Изображение картины можно получить не только на экране: цветной принтер переносит его на бумагу. Репродукции пока не идеальны, но уже превосходят по качеству многие типографские. Так что изготовление каталогов для будущих выставок тоже ляжет «на плечи» ЭВМ.

— В дальнейшем банк данных вместит информацию обо всем русском и советском искусстве музеев Советского Союза, — говорит генеральный директор Всесоюзного музейного объединения «Государственная Третьяковская галерея» Ю. К. Королев. — Пока мы с вами еще даже не знаем, каким богатством обладаем. Такая систематизация позволит нам прежде всего получить исчерпывающую информацию обо всех произведениях, которые хранятся в музейных

собраниях — больших и малых. Это расширит возможности для организации различных художественных выставок: можно будет привлекать для этого фонды всех музеев страны.

Кстати, в Национальной художественной галерее в Вашингтоне подобная работа уже

проделана. Изображения более 1,5 тыс. экспонатов вместе с пояснительными текстами записаны на оптический диск, нужные данные на котором компьютер находит за доли секунды. Пожелаем отделу информатики Третьяковки иметь у себя и такие диски!

## После конкурса

Итогом конкурса на лучший учебник по информатике для учителей стали три рукописи. Две, занявшие третье место, рекомендованы к изданию в качестве книг для чтения, а рукопись, занявшая второе место (первое не присуждалось), — в качестве пробного учебного пособия и для использования в педагогическом эксперименте после переработки. Подробный анализ недочетов и недостатков этой рукописи приведен в статье старшего научного сотрудника НИИ СиМО В. К. Белашапки, опубликованной в журнале «Информатика и образование» (1988, № 1).

Как и другие авторы, а изъявили желание участвовать в конкурсе более 20 авторских коллективов, мы хорошо понимали, что в поставленные сроки — за 1 год — подготовить рукопись в соответствии с конкурсной программой архисложная задача. Не случайно большая часть авторов не представила рукописей на конкурс. Наш авторский коллектив\* в силу ряда обстоятельств приступил к написанию рукописи лишь за три месяца до начала конкурса. Но мы все же предпочли представить черновой вариант. Основной расчет был на издание рукописи в качестве книги для чтения и ее использование для проведения эксперимента в средних школах. Итог конкурса для нас оказался неожиданным — победное второе место и возможность подготовки пробного учебного пособия.

Что привлекло конкурсную комиссию к этой рукописи? Судя по представленным рецензиям, главные причины — преимствен-

ность к действующему учебнику, систематическое применение ЭВМ, современный взгляд на информатику и программирование, а также ориентация на конкурсную программу. Преимственность рукописи к действующему учебнику заключается в сохранении в основном его структуры и использовании алгоритмического языка как основного методического средства обучения началам программирования. Алгоритмический язык и в рукописи, и в учебной практике авторов применяется в качестве средства описания логики работы программ для ЭВМ. По методике авторов ЭВМ предполагается использовать с первого до последнего урока, а в качестве языка программирования излагается Бейсик в тесной взаимосвязи с алгоритмическим языком. Апробация материалов рукописи показала возможность их применения на ЭВМ «Корвет», «Электроника-УКНЦ», БК-0010Ш и даже на БК-0010 с языком Фокал путем замены одной единственной главы.

Главными методическими целями представленной рукописи строго в соответствии с требованиями конкурсной программы являются овладение всеми учащимися компьютерной грамотностью и формирование у них информационной культуры. Компьютерная грамотность трактуется как умение читать и писать, считать и рисовать, работать с информацией и программами на ЭВМ. Эта цель достигается за 30—40 часов занятий с учащимися старше 12 лет. Информационная культура понимается строго по конкурсной программе: 1) навыки грамотной постановки задач; 2) представление о подборе формали-

\*В. А. Каймин, А. Г. Щеголев, А. В. Гиглавый, Е. А. Ерохина, Д. П. Федюшин.

зованных моделей и методов решения задач; 3) умение применять алгоритмический язык для описания алгоритмов решения задач; 4) умение составлять простейшие программы и навыки их отладки на ЭВМ; 5) знание основных возможностей и устройств ЭВМ; 6) умение интерпретировать результаты, получаемые от компьютеров.

Единственным отклонением рукописи от конкурсной программы было применение новых систематических методов составления и анализа алгоритмов и программ, позволяющих учителям быстро проверять их правильность, выявлять и исправлять вкравшиеся ошибки. Данная проблема легко решается, во-первых, при описании алгоритмов (до составления программ) именно на изучаемом сейчас в школах алгоритмическом языке, в который добавлены операции ввода — вывода, и во-вторых, при обязательной записи постановок задач, строгом описании методов их решения и сценариев работы с ЭВМ. В настоящее время отсутствуют зарубежные прототипы в изложении этой методики. Не совсем удачно изложение техники доказательных рассуждений, с помощью которых проводится анализ правильности алгоритмов и из-за которых в курсе информатики становится необходимым изучать элементы логики, отмечено В. К. Белошапкой как главный недостаток рукописи. В то же время мы обнаружили, что именно эта техника формирует навыки грамотных постановок задач, тщательного планирования действий и умения логически обосновать предлагаемые планы их решения, т. е. способствует достижению целей, которые могут и должны стать педагогической основой информатики как учебного предмета.

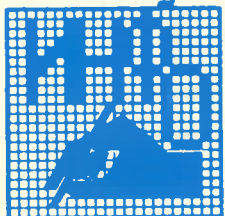
Поставленные цели заставляют по-новому взглянуть на структуру и содержание учебника. Развитие логического мышления требует вводить обучение элементам логики как можно раньше, до изучения основ алгоритмизации и программирования. При этом необходим хорошо продуманный практикум. Таким практикумом может стать решение на ЭВМ задач логического характера с исполь-

зованием известного языка логического программирования Пролог. Основы алгоритмизации можно и нужно изучать, не только реализуя алгоритмы в форме программ на ЭВМ, но и вводя технику логической оценки результатов их работы. И наконец, завершением этой логической линии в учебнике информатики должно стать изучение методов составления алгоритмов решения задач с одновременным доказательством отсутствия в них ошибок, так чтобы все построения заканчивались словами «что и требовалось доказать». Именно эта линия и была заложена в представленную на конкурс рукопись учебника информатики.

В каком состоянии сейчас пробное учебное пособие? В сентябре 1987 г. при поддержке Главного управления народного образования и Московского института усовершенствования учителей начался педагогический эксперимент в ряде средних школ Москвы. Одновременно осуществляется методическая переработка учебника в соответствии с замечаниями рецензентов конкурсной комиссии и предложениями учителей и методистов, участвующих в эксперименте. В настоящее время третья редакция рукописи учебного пособия сдана в издательство «Просвещение». Совместно с ведущими московскими методистами и учителями-практиками готовится методическое пособие и задачник по информатике. Предварительное знакомство с методикой и материалами издаваемого учебного пособия можно получить в следующих номерах журнала «Информатика и образование», а также в журналах «Квант» (1986, № 10, 11, 12; 1987, № 6), «Вопросы психологии» (1986, № 6), в сборнике «Проблемы компьютерного обучения» (1986, вып. 1), в учебных пособиях «Основы информатики и вычислительной техники» (МИЭМ, 1985), изданном для учителей во время первого компьютерного всеобуча, и «Основы доказательного программирования» (МИЭМ, 1987).

**В. А. КАЙМИН.**

*Доцент, кандидат технических наук*



## Компьютеризация в школе и развитие самостоятельного мышления у учащихся

Я услышал и забыл.  
Я увидел и запомнил.  
Я сделал и понял.

(Девиз  
школы Беббиджа)

Проблемы компьютеризации школьного образования в настоящее время являются предметом исследования многих специалистов, однако единой и общепринятой концепции до сих пор не существует. Введение же нового курса является лишь первым этапом компьютеризации и ни в коей мере не должно ограничиваться овладением учащимися новой деятельностью — алгоритмизацией и программированием на компьютере. Это становится ясно уже большинству специалистов, непосредственно не имеющих дело с системой школьного образования (см.: ЭВМ в школе — реальность наших дней: «Круглый стол» журнала ЭКО. Наука. 1984. № 11). Однако по-прежнему на страницах журнала «Информатика и образование» около 70 % всех материалов составляют статьи, содержащие рекомендации, прямые указания именно по преподаванию курса на базе пробного пособия. Тем самым наращивается фундамент, который не позволит в будущем предложить курс, более отвечающий требованиям времени (или даже реализовать все положительные идеи, которые были первоначально заложены в существующий курс и извращены на этапе перехода к массовому внедрению в школы). Сравнительно редко в журнале помещаются мнения, отличные от признанного Минпросом СССР, например раздел «Точка зрения» в вашем журнале (№ 5 за 1987 г.). Доля таких публикаций не превышает 10 %, и это в наше время, когда необходимо «овладевать культурой дискуссии, уважительно относиться даже к крайним точкам зрения для поиска путей решения общей цели».

Не нужно бояться открытых дискуссий и опасаться, что учителя что-то «неправильно» поймут либо выработают свое собственное мнение. Основная беда нашего среднего образования (да и не только образования) и состоит именно в нежелании, боязни учителей работать самостоятель-

но, без указки сверху, без разжевывания материала до атомарной консистенции. И при этом мы еще хотим, чтобы наши ученики обладали самостоятельным мышлением и овладевали материалом на уровне содержательного, творческого обобщения. Настало время понять, что методы насаждения готовых рецептов по всем вопросам компьютеризации, культивируемые при директивно-централизованной системе управления школьным образованием, никак не способствуют достижению конечной цели обучения — формированию квалифицированного пользователя средств вычислительной техники, обладающего творческим, научно-теоретическим мировоззрением и в совершенстве овладевшим основами формальных методов познания, или, проще говоря, «подготовить школьников к жизни в информационном обществе». (Информатика и образование. 1987. № 5. С. 113).

Для развития навыков самостоятельного мышления прежде всего необходимо широкое знакомство учащихся с «первоисточниками», т. е. с оригинальной научно-технической литературой и документацией по компьютерной тематике. Уместно вспомнить хотя бы тот факт, что любой учебник по конкретной дисциплине для педагогических вузов резко отличается от своих «собратьев» — учебников, предназначенных для непедагогических институтов. И при этом ни для кого не секрет, что «дети усваивают информатику лучше взрослых, а взрослые хуже детей» (см.: там же. 1987. № 5. С. 117).

В условиях, когда появляется все больше доступной литературы по различным вопросам использования компьютеров, по крайней мере неразумно закрывать глаза на то, что решение многих задач из пособия, особенно у учащихся обычных школ, способствует потере всякого интереса к использованию компьютеров в своей будущей деятельности. Безусловно, не следует заигрывать с учеником и потакать его лени, однако первый опыт преподавания нового курса говорит о резком повышении заинтересованности детей учебой. Для поддержания такого энтузиазма надо дать возможность школьникам самостоятельно выбрать себе

задачи, ставить перед собой проблему и искать пути ее решения. Преподаватель только ориентирует ученика в море информации, дает советы и помогает выбрать наилучший вариант решения.

Теперь несколько слов о программном обеспечении курса. Нельзя забывать, что одним из преимуществ, ожидаемом от широкого применения персональных компьютеров в школе, является возможность организации индивидуальной работы, что заложено даже в самом названии «персональный», а любая массовость может привести к негативным последствиям. Не всякий учитель может перенять опыт у своего коллеги; так же и с компьютерными программами: не всякой школе нужны именно те или иные программы. При централизованном распространении программ желательно стремиться только к совместимости используемых компьютеров и к унификации их операционных систем. Большой же интерес у учителя могут вызвать программы-инструментарии, представляющие собой некую заготовку, форму, «заполняя» которую своим собственным накопленным опытом, учитель получает конечный програм-

мный продукт, отвечающий требованиям конкретных условий обучения. Такими программами-заготовками могут быть графические модули, игровые блоки, музыкальные программы и т. п. Все эти программы должны быть открытыми (т. е. позволять легко встраивать свои собственные модули), хорошо документированными и поставляться вместе с текстами.

В заключение еще раз заметим, что нельзя подходить к изучению информатики со старыми приемами обучения обычным школьным предметам. Сама по себе информатика не уместается в рамки традиционного предмета, а пронизывает все школьные курсы как естественного, так и гуманитарного циклов. Как следствие этого, необходимы другие методики преподавания, подбора задач, необходима новая система контроля усвоения знаний и т. п. В противном случае, как это уже неоднократно было, все потраченные усилия окажутся напрасными, а материальные средства не дадут отдачи.

Д. ВОЛКОВ

117

## Об алгоритмической нотации

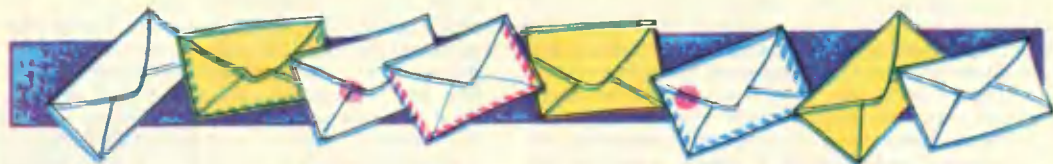
Как известно, в 1985 г. вышла в свет I часть пробного учебного пособия «Основы информатики и вычислительной техники» под редакцией А. П. Ершова и В. М. Монахова. В этой книге, в соответствии с концепцией надязыкового подхода к обучению программированию, был предложен некий алгоритмический язык, вокруг которого ведется немало споров среди учителей информатики (особенно имеющих в распоряжении вычислительную технику). Можно было бы устранить почву для этих споров, предложив иное название: алгоритмическая нотация, используемая для изображения алгоритмов без учета особенностей реальных языков программирования. И средство это может быть более мощным, оставаясь доступным для учащихся, более наглядным, нежели усеченный вариант, приведенный в учебном пособии (по непонятным причинам отсутствует цикл до, да и комментарии не помешали бы).

— Из-за присутствия в наименовании слова «язык» алгоритмическая нотация (из самых лучших побуждений) была реализована на компьютере (Е-практикум). А ведь это примерно то же, что делать машинную реализацию блок-схем, также служащих для изображения алгоритмов.

Читателю, желающему более подробно ознакомиться со всеми достоинствами алгоритмической нотации высокого уровня, можно предложить монографию Б. Мейера и К. Бодуена «Методы программирования» (в двух томах), авторы которой «далеки от мысли прибавить еще один новый язык к нескольким сотням, имеющимся уже на «рынке!»»

М. МАТВЕЕВ

От редакции. Мнение о ненужности языка программирования с русской лексикой еще довольно распространено, хотя полезность такого языка можно считать вполне доказанной. Недаром он существует во множестве вариантов — от Е-практикума на «Ямахе» до языка персональных вычислений на ЕС ЭВМ. Вот только как его (их?) называть? Предложение М. Матвеева о переименовании безмашинного алгоритмического языка в алгоритмическую нотацию представляется обобщенным, а предложений о названии соответствующего языка программирования мы ждем от вас, дорогие читатели.



## ЖУРНАЛ И ЧИТАТЕЛЬ

Каждый день приходят в редакцию новые письма. С радостью воспринимаем мы хорошие слова в адрес «ИНФО», со вниманием критику. Баланс «позитива» и «негатива» будем стараться выдерживать в публикации строк из писем. А теперь поговорим о том, что волнует наших читателей.

118

На первое место по числу жалоб выходит голод на литературу по компьютерной тематике, каждое второе письмо об этом. Сюда же примыкают письма с предложениями о развитии нашего журнала. Учитывая, что такое издание как наше — единственное в своем роде в стране — читатели предлагают сделать журнал ежемесячным (студент из Хабаровска Г. Векслер, преподаватель из Кызыл-Орды А. Крупин и др.), обязательно создать приложение к журналу (студент из Белгорода М. Колесников и др.). Есть просьбы обязательно продолжить рубрику «У истоков» (Г. Меценко, г. Северск), «Молодежная инициатива» (десятки писем).

Коль скоро мы перешли к содержанию журнала, следует сказать, что многие письма-отклики на те или иные статьи содержат подробный разбор публикаций, что очень помогает редакции. Рекордсменом по количеству положительных отзывов стала статья А. Архангельского «Мир ЭВМ» (редактор К. Шеховцев). Многие читатели просят продолжить эту тему, что мы с удовольствием делаем в этом номере. Много споров вызвала статья Т. Поддубной «Информатика по опорным сигналам». В ряде коллективных учительских писем выражается восторженное мнение об этой публикации. Но есть и критические суждения. Вот что пишет доктор педагогических наук, профессор А. Столяр из Могилева: «Информатика полна собственных опорных схем и нет никакой надобности в нагромождении одних сигналов на другие». Отметим, что в целом, по мнению автора последнего письма, журнал наш очень актуальный и полезный, а его высказывание продиктовано стремлением «уберечь информатику от всякого рода наслоений, хотя бы на период формирования этого нового учебного предмета».

Публикуя статью Т. Поддубной, редакция преследовала цель, не навязывая учителям «истину в последней инстанции», показать один из методических подходов к подаче учебного материала. Право современного учителя — выбрать то, что ему ближе и отсечь то, что не годится.

Нам приятно отметить, что у журнала появились новые группы подписчиков — из писем следует, что «Информатика и образование»

привлекла внимание инженеров, школьников, программистов.

Спасибо всем, приславшим в редакцию письма. Ваши мнения, уважаемые читатели, будут обязательно учтены в работе над журналом.

## СТРОКИ ИЗ ПИСЕМ

Ажиотаж, поднятый по поводу внедрения КУВТ в школах, мгновенно сменяется разочарованием, если прекратить использовать эту дорогостоящую технику в качестве игровых автоматов.

Можно и на линкорке выезжать на рыбалку, но рыба уже будет «золотой».

Школы, не имея грамотных кадров, не имея программного обеспечения и методик, поневоле используют первое, чтобы привлечь ребят, — игры. А ведь наша цель — компьютерная грамотность.

Мое мнение — пока мы еще не так богаты — сконцентрировать КУВТы в межшкольных учебно-производственных комбинатах, где легче организовать эксплуатацию, обеспечить программами и наладить техническое обслуживание. УПК должен быть и методическим центром для учителей.

**В. СЕЛУЯНОВ,**  
преподаватель информатики  
г. Сочи

В экспериментальном учебном плане предмет «Основы информатики и вычислительной техники» перенесен в VI—VII классы с целью применения полученных вычислительных навыков при изучении других предметов. О каких полученных навыках можно говорить? В области азартных игр? Но это не компьютерная грамотность! Чтобы ученики начали думать, сами создавать программы, они должны иметь математическую культуру. Что изучается по математике в V—VII классах? Натуральные числа, рациональные числа, действия над десятичными, обыкновенными дробями, линейные уравнения, одночлены, многочлены. Честно говоря, маловато для преподавания информатики. Ведь в этом курсе мы обучаем алгоритмическому мышлению, структурному подходу к предмету. Ученики в VI—VII классах не знают квадратных уравнений, рациональных уравнений, линейных неравенств и систем, арифметической, геометрической прогрессии, тригонометрии. И возникает вопрос: что уче-



ники будут делать с компьютерной техникой, если не играть?

Давайте не путать двух вещей — компьютер-игрушка и компьютер — техническое средство для овладения разными предметами в средней школе.

Не сомневаюсь, что вы знаете об объявленном конкурсе на новый учебник «Основы информатики и вычислительной техники». Программа курса рассчитана на обучение основам информатики в IX—X классах. Предполагается, что новый учебник охватит высокие материи на уровне высшей школы, а курс информатики будет перенесен в VI—VII классы. Ученые шутят?

С уважением

**Ю. АМБРАСАС**  
ст. преподаватель  
факультета математики  
Вильнюсского  
университета

Взяться за письмо меня заставило то состояние, в котором сейчас пребывают преподаватели «из глубинки».

Ваш журнал я выписываю с самого первого номера. Не могу не отметить, что многие, очень многие статьи были для меня очень полезны.

Теперь о главном. Два «лета» подряд я по две недели был на курсах. Вы думаете, они мне что-то дали для работы? Нет. Я отдаю должное учителям курсов г. Волгограда, преподавателям ВГПИ, у которых я когда-то учился, они делали очень многое. Но... Но на курсах обучались учителя из сел решению задач /элементарных/ из курса информатики. И, что было для меня чудовищно дико, многие из них /конечно, эти села отдаленных районов/, не понимают команды  $x:=x+1$ , уже 2 года /!?!/ ведя предмет.

**В. ПЕТРОВ**, преподаватель зооветтехникума  
г. Дубовка Волгоградской обл.

Чуть не плача от досады, смотрел на днях передачу «Спокойной ночи, малыши», когда увидел, что в каком-то детском саду есть компьютеры.

И думается: мне бы! Но... Учитель не хозяйственник, и где бы /на различных уровнях/ я ни пытался заводить разговор об оборудовании в нашей школе кабинета информатики, меня в лучшем случае вежливо выслушивали — не более того. О худших случаях и не хочется говорить.

**С. СТЕПИН**  
учитель информатики, г. Новосибирск

Из доброжелательного в целом письма:

Обратите, пожалуйста, внимание на опечатки. Досадно замечать даже пропущенные или лишние запятые, а уж «смысловые» ошибки в издании учебного характера просто недопустимы, тем более в издании по информатике: ведь первый же урок для человека, садящегося за терминал, превращается в урок аккуратности...

...При современном уровне «компьютерной грамотности» говорить о внедрении ЭВМ в повседневную жизнь просто нелегко. Поэтому одна из насущных, на мой взгляд, задач нашего подразделения — нескольких программистов, собравшихся вокруг ЭВМ, — это популяризаторство, просветительство, агитация — называйте, как хотите. Иначе мы рискуем остаться в башне из ЭВМ, как дань моде, а вокруг будет идти жизнь проектного института, как шла и десять, и чуть не сто лет назад.

Было бы очень хорошо, если бы слово «образование», вынесенное в заголовок вашего журнала, не сводилось к школе. Когда речь идет о самом настоящем «компьютерном ликбезе», кто бы посоветовал, что и как говорить взрослым, а то и почтенного возраста людям, чтобы они поняли хоть что-нибудь сверх того, что «машина считает»!

**В.В.Ш.**

/подлинный адрес и фамилия редакции известны/

...Интересны работы учащихся, т. к. именно они нагляднее всего демонстрируют сегодняшние достижения массовой школы в обучении информатике.

Учителям интересны вопросы, связанные со школьной техникой, оборудованием кабинета, санитарно-гигиенические требования.

Не остаются без внимания материалы о проблемах и перспективах компьютеризации школы. Всем начавшим преподавание с безмашинного варианта курса ОИВТ нужно знать, что ожидается завтра, какая техника придет в школу, как нужно готовиться к ее использованию. Это помогает более активно вести сегодняшнюю работу.

Журнал читают не только учителя, но и ученики. Рубрика «Молодежная инициатива» способствует повышению интереса к предмету.

Очень жаль, что пока этого журнала нет в обязательной подписке школы.

Очень важны в журнале не только статьи, но и иллюстрации — подчас у сельского учителя это пока единственная возможность показать детям дисплей, клавиатуру, микропроцессор, БИС. Качество иллюстраций сегодняшнего журнала оставляет желать лучшего. Особенно плоха обложка. После работы с журналом за две-три недели не только от картинок не остается следа, но и от самой обложки. А журнал — постоянная рабочая литература учителя, и документ, и наглядное пособие. Как же его сохранить для постоянной работы с ним? По-видимому, издательство недостаточно понимает роль и значение журнала в становлении нового направления развития нашей школы.

Назрела необходимость организовать выпуск приложения, в котором помещался бы методический наглядный материал, алгоритмы и программы школьников, профессиональные программы для школы и т. п., т. к. потребность в размножении такого рода материалов очень велика.

**Н. АЛФЕРЕНКО, В. БОРИСОВ,**  
**В. ЗАРЬЯНЦЕВА, А. МАКСИМОВ**, учителя  
информатики, г. Саратов  
**Л. ГОЛДА**, учитель информатики, г. Энгельс

## «Полиграфбуммаш-87» глазами специалистов

120

В Москве с 23 по 30 ноября 1987 г. проходила международная специализированная выставка «Машины и оборудование для полиграфической и бумажной промышленности» — «Полиграфбуммаш-87». На выставке было представлено современное оборудование ведущих фирм в области компьютеризированного выпуска книжной и журнальной продукции: «Сканграфик», «Бертольд», «Хелл», «Линотайп» (ФРГ), «Компьюграфик» (США), «Кроссфилд», «Монотайп», «Ренк ксерокс» (Великобритания), «Дайниппон скрин» (Япония), «ХТС» (Швеция), «Хайфен» (Италия), «Норск дейта» (Норвегия).

Анализ представленных экспонатов и проспектов новых разработок зарубежных фирм позволяет классифицировать современные системы по автоматизированной обработке текста и иллюстраций.

К *первой группе* можно отнести системы с централизованной обработкой и хранением массивов данных под управлением центрального процессора, реализованного на базе мини-ЭВМ типа PDP-11, с развитой периферией рабочих мест.

Системы осуществляют набор и редактирование текста, подготовку логотипов, верстку текста (компоновку элементов текста на страницы издания в соответствии с требованиями полиграфии), вывод массивов сверстанных данных на фотонаборные автоматы для получения диапозитивов полос. В процессе верстки для иллюстраций обычно оставляются окна, но в некоторых системах есть рабочие места для геометрической обработки иллюстраций.

Типична для этой группы система серии D фирмы «Бертольд», предназначенная для производства книжной продукции. Управление процессом набора, верстки текста и накопления данных в серии D осуществляется центральным процессором системы dms 5000 с емкостью внешней памяти 2×70М байт (улучшенный вариант серии — dms 7000 с емкостью 2×140М байт). Центральный процессор обеспечивает подключение до 5 терминалов, среди которых фотонаборный автомат на ЭЛТ дру 5000, позволяющий одновременно использовать до 200 гарнитур при ско-

рости вывода на носитель 280 000 знаков/ч. Для ввода и редактирования текста фирма предлагает персональные компьютеры PC—Berthold, совместимые с IBM PC XT, на которых используется программное обеспечение фирмы «Текстино-+», ориентированное на работу с несколькими языками, в том числе и русским. Сложный многоколонный набор, таблицы, формулы вводятся со специального двухэкранного терминала gst 7003: на одном (малом) отображается текст с типографскими командами, на другом — результирующее представление страницы текста. Терминал снабжен расширенной клавиатурой для сложного набора.

Верстальный терминал tps 7001 также оформлен в виде двухэкранного рабочего места. Верстка обеспечивает компоновку текстов, набранных на персональных компьютерах и на gst 7003. В процессе верстки пользователь может изменять кегль (размер) основного текста и заголовков, оформлять страницу в виде многоколонного набора, изменять размеры окна для иллюстраций, просматривать результаты своих действий на большом экране в виде, приближенном к типографскому.

Корректурные отиски верстки изготавливаются на системном принтере на обычной бумаге. Для издательств и типографий СССР фирма подготовила набор русских шрифтов, программу переноса согласно русской грамматике с использованием словаря исключений, провела работу по обеспечению функционирования фотонаборного автомата на советской фотопленке и фотоматериалах. Для расширения функций системы D в сфере редакционно-издательских работ фирма предлагает вариант установки компьютеров, снабженных редактором текста ТЕХТИНО+, непосредственно в редакциях. Обмен информацией с типографской фотонаборной системой D осуществляется через дискеты. Операционная система PC—Berthold допускает использование других редакторов текста или разработку собственных программных средств.

Если фирма «Бертольд» предоставляет своим пользователям для сложного многоколонного набора специализированные терминалы с расширенной клавиатурой, то фирмы «Хайфен» и «Моно-

тайп» демонстрировали программу сложного набора на персональном компьютере. Разработанное фирмой «Хайфен» программное обеспечение позволило значительно расширить возможности персонального компьютера, включив функции выбора типа переноса (автоматического словного и со слогоделением), предоставить пользователю систему «меню» на русском языке, ввести некоторые элементы верстки с отображением результатов на экране персонального компьютера.

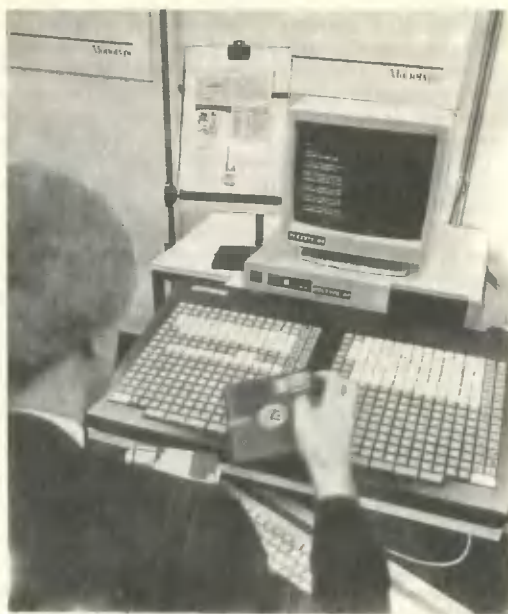
Для набора математических формул фирма демонстрировала терминал POLTYPE 04, позволяющий создавать специализированную систему из нескольких терминалов с получением результатов верстки на лазерном принтере и с выходом одного из терминалов системы на фотонаборный автомат.

Ко второй группе систем автоматизированной обработки текста и иллюстраций можно отнести настольные типографии (Desk-top Publishing). Они предназначены для выпуска изданий небольшими тиражами при невысоких требованиях к качеству печатной продукции.

В комплект настольной типографии входят компьютер типа IBM PC AT со стандартной клавиатурой, плоский сканер для ввода штриховых и полутоновых иллюстраций и лазерный принтер для получения готовой продукции или вывода сверстанной полосы на прозрачный носитель, используемый для получения печатной формы. Эти системы устанавливаются непосредственно в редакциях (издательствах) и позволяют осуществлять весь технологический процесс подготовки издания к выпуску на одном рабочем месте. Программное обеспечение приспособлено к проведению всех операций многоколонного набора текста, редактированию, геометрической обработке иллюстраций, верстке полосы текста. Из систем этого класса на выставке была представлена система фирмы «Ренкскерокс» с программным обеспечением Ventura.

Системы этой группы отличаются малыми размерами, невысокой стоимостью, дружелюбным интерфейсом пользователя, широким набором программных средств, используемых для редактирования текста и верстки изданий, достаточно большим объемом внешней памяти.

Невысокая разрешающая способность сканеров и сложность получения качественных фотоформ



Терминал POLTYPE 04 фотонаборной системы фирмы «Монотайп».

на лазерных принтерах пока еще сдерживают широкое применение данных систем для выпуска книжной и журнальной продукции.

Третья группа систем в полном комплекте на выставке не была представлена, хотя анализ проспектов новых разработок фирм «Бертольд», «Компьюграфик», «Сканграфик», «Линотайп» и др. позволяет классифицировать системы как децентрализованные, построенные на базе автоматизированных рабочих мест, ориентированных на автономное выполнение одной из технологических процедур процесса выпуска изданий и соединенных между собой в вычислительную сеть для приема-передачи результата на следующую технологическую процедуру.

Все системы этого класса предназначены для интегрированной обработки текста и иллюстраций, в них решены задачи цельнополосной верстки и



Автоматизированная фотонаборная установка Skantext-1000 фирмы «Сканграфик» (ФРГ).

подготовки макетов печатных листов на рабочих графических станциях. Системы включают автоматизированные рабочие места со сканерами для черно-белых штриховых и полутоновых иллюстраций с развитым графическим редактором, позволяющим осуществлять кадрирование, масштабирование и электронную ретушь изображений. Для обработки цветных изображений применяются устройства с цветodelением и его последующей коррекцией на экране.

Примером такой системы является система серии М фирмы «Бертольд», в состав которой входят сканер для обработки штриховых и полутоновых оригиналов, блок памяти для хранения иллюстраций, рабочие станции для набора и редактирования текста, графическая рабочая станция для цельнополосной верстки, лазерное печатающее устройство и фотонаборные автоматы. Графическая станция Graphic Workstation с размером экрана по диагонали 480 мм и разрешающей способностью 900×1152 оснащена компьютером с 32-разрядным процессором типа М68020 (объем ОЗУ — 4М байта). Для управления перемещением курсора используется устройство типа «мышь».

122

Иллюстрации на экране воспроизводятся с достаточно грубой разрешающей способностью, обеспечивающей обработку их геометрических размеров и вывод результатов верстки полосы на лазерном принтере для получения корректурных оттисков. Результаты геометрической обработки иллюстраций «учитываются» при выводе иллюстрационных массивов данных на фотоматериал. Все рабочие места объединены единой локальной сетью Ethernet. Аналогичные рабочие станции разработаны фирмами «Компьюграфик» и «Линотайп».

В завершение обзора систем автоматизированной обработки текстовой и иллюстративной информации, представленных на выставке «Полиграфбуммаш-87», можно отметить следующее.

#### Лазерное устройство фирмы «Монотайп».



Система Хромоком фирмы «Хелл».

Все фирмы имеют тенденцию к интегрированной обработке текста и иллюстраций.

Системы строятся по модульному принципу, позволяющему удовлетворять различным пользовательским запросам: от малогабаритных издательских до высокоразвитых вычислительных систем, устанавливаемых в типографиях, издательствах и издательско-полиграфических объединениях. При этом процесс компьютеризированной обработки издания все больше сдвигается в сторону редакционно-издательской деятельности.

Широкое внедрение персональных компьютеров на различных этапах подготовки и выпуска печатной продукции позволяет расширить круг пользователей, как включенных в общую вычислительную сеть, так и функционирующих автономно.

Почти все фирмы предоставили программы для работы наборщиков и редакторов на русском языке, включая программы слогаделения, словарной проверки, а также ограниченный набор русских шрифтов. Однако далеко не все фирмы предусмотрели возможность использования советских фотоматериалов, особенно при работе на фотоавтоматах с лазерным принципом действия.

К сожалению, на выставке «Полиграфбуммаш-87» не были представлены отечественные системы обработки текстовой и иллюстративной информации. И неудивительно. Отечественная фотонаборная техника, используемая в настоящее время (системы «Каскад», ДИС), а также технические решения, закладываемые в новые разработки, не всегда отвечают требованиям сегодняшнего дня.

Хочется верить, что проведенная выставка окажет существенную помощь в разработке отечественных систем, даст импульс инженерной мысли.

А. ПАРХАЧЕВ  
Г. ШЕЙНА

# СССР — Болгария: сотрудничество в области образования

В январе 1988 г. Академия наук СССР приняла представительную делегацию болгарских ученых, занимающихся созданием новых систем школьного образования.

Коллектив, который представляли наши коллеги, существует почти девять лет. В июле 1979 г. в Болгарии был принят закон о реформе народного образования, предусматривавший постепенный переход к реализации идеи школы продленного дня. Двенадцатилетний период обучения начинается с шести лет. Главная задача реформы — повышение качества работы школы, подъем уровня знаний выпускников. Но программные документы лишь наметили основные пути реализации этой задачи и не содержали конкретных инструкций по изменению содержания образования, самого педагогического процесса. В стране начались поиски путей обновления школы.

При Академии наук Болгарии была создана Проблемная группа по образованию (ПГО), которую возглавил академик Благовест Сендов. Ученые решили радикально пересмотреть и содержание школьных курсов, и методы обучения, целиком подчинив их требованиям времени. Небольшая группа университетских профессоров (два математика, два лингвиста и психолог) на несколько недель заступили на почти круглосуточную вахту по подготовке учебника для шестилеток.

Уже в 1979/80 учебном году в трех классах (в Софии, в окружном городе Благоевград и в одной из деревень) первоклассники получили экспериментальный учебник. Первые результаты обнадежили, и на следующий год число экспериментальных школ увеличилось. Сейчас по программам ПГО занимается около 13 тыс. учащихся в 28 школах Болгарии. За восемь лет Проблемная группа по образованию издала для ребят и учителей около 150 книг. Идеи ПГО получили серьезную проверку временем.

Что же отличает педагогическую концепцию болгарских ученых? Основная идея — интеграция различных школьных дисциплин и их информатизация. В начальной школе один главный предмет — «Читаем, пишем, считаем»; с V класса основные знания и навыки дети получают в четырех интегрированных курсах: «Природа», «Общество», «Язык и математика», «Производство». Каждый из этих курсов учитывает идеи информатики, но наиболее серьезно она преподается в курсе «Язык и математика». Обучение ведется на базе языка Лого с использованием компьютеров «Правец».

Удался ли эксперимент? Судите сами. Знаний у ребят в экспериментальной школе не меньше, чем в обычной. Они получают солидную эстетическую подготовку. Каждый день у них урок физкультуры — они крепче своих ровесников из других школ. А еще не бывает домашних заданий и нет занятий по субботам. В общем, ребята и учителя довольны.

Неудивительно, что у нас в стране заинтересо-

вались болгарским опытом. В 1986 г. образовалась группа ученых-энтузиастов под руководством академика Е. П. Велихова, которая со временем обрела и организационное единство, составив временный научно-технический коллектив «Школа-1». В составе коллектива специалисты по самым разным отраслям знаний — математики и психологи, биологи и лингвисты. Советские ученые, как и их болгарские коллеги, будущее образования видят в интеграции и координации всех школьных дисциплин на базе новых информационных технологий, придают большое значение не только разработке курса информатики, но и информатизации школы в целом, подготовке педагогических программных продуктов по всем школьным дисциплинам, а в связи с этим и пересмотру содержания образования и педагогических приемов. Конечно, особое внимание уделяется начальной школе, где наряду с основами других знаний должна даваться преподавателю информатики.

Сходство целей советской и болгарской групп привело к тесному сотрудничеству. Основным пунктом программы январского визита коллег по ПГО был трехдневный советско-болгарский семинар по новым информационным технологиям в образовании.

Научное руководство семинаром осуществляли вице-президент АН СССР академик Е. П. Велихов и вице-президент АН Болгарии академик Б. Сендов.

Первый день работы семинара был посвящен программированию. Е. Сендова рассказала о большом школьном курсе информатики, о реализации педагогических идей, лежащих в основе языка Лого в курсе геометрии. Т. Е. Семенова изложила программу работ по использованию в педагогических целях систем представления знаний. С. Ф. Сопрунов сообщил о педагогических возможностях нового расширения языка Лого — Логорайтера, интенсивная работа с которым ведется в Москве с декабря 1987 г. после визита в СССР автора Лого и Логорайтера — С. Пейперта (см.: Информатика и образование. 1988. № 2).

А. Г. Кушниренко дал подробную характеристику программному обеспечению нового школьного курса информатики и Логорайтера. Разрабатываемым в ВНТК «Школа-1» педагогическим программным продуктам различного назначения были посвящены доклады М. А. Ройтберга, Е. А. Диница, Г. В. Калибердиной.

Второй день начался с посещения средней школы № 57 — одной из школ-лабораторий временного коллектива. Здесь болгарские коллеги познакомились с практической реализацией некоторых идей сотрудников ВНТК «Школа-1. Урок истории в IV классе одновременно вели историк В. Р. Лецинер и математик С. Г. Смирнов. Главную задачу они видят в обучении ребят логике процесса исторического развития, системному подходу к истории. В V классе учащихся ждут уроки истории с использованием компьютера. Большой интерес у болгарских ученых вызвала специальная углубленная программа по математике, по которой ведутся заня-

тия в старших классах школы. Гости из НРБ посетили урок математики Р. К. Гордина и семинарское занятие А. Шеня.

В этот день участники советско-болгарского семинара заслушали также доклады Н. Я. Виленкина о построении школьного курса математики, А. М. Абрамова о работе с одаренными учащимися, А. К. Поливановой о разрабатываемом курсе для I класса, Д. Лазарова и Е. Головинского о структуре интегрированного курса «Природа» и связанных с ним факультативах, Е. Сендовой о месте математических занятий в курсе «Язык и математика», В. И. Беликова и А. К. Поливановой о построении языковых курсов и взаимосвязи лингвистики и информатики.

Заключительный день семинара открылся пленарным заседанием, на котором Б. Пенков и А. Л. Семенов подробно охарактеризовали историю и современное состояние работ в ПГО и ВНТК «Школа-1»; сотрудники временного кол-

лектива С. Ф. Сопрунов и В. Е. Юрковский ознакомили участников семинара с педагогическими возможностями управляемого компьютером робота. Доклад К. Манова был посвящен курсу «Производство» и трудовому обучению в экспериментальной болгарской школе. Л. Б. Переверзев рассказал об организации трудового обучения и профориентации в школах-лабораториях временного коллектива и о дальнейших планах.

В ходе семинара разработаны детальные планы сотрудничества между ПГО АН Болгарии и ВНТК «Школа-1» АН СССР в области создания учебных пособий по информатике, языку и математике, интегрированному курсу «Природа» и по другим направлениям. Советско-болгарское сотрудничество по новым информационным технологиям в образовании принимает все более регулярный и плодотворный характер.

**В. БЕЛИКОВ,  
А. СЕМЕНОВ**

## Семинар в Воронеже

В январе 1988 г. Министерство высшего и среднего специального образования РСФСР провело в Воронеже совещание-семинар. Сюда приехали председатели методических объединений преподавателей основ информатики и вычислительной техники средних специальных учебных заведений России. Всего собралось около ста человек. Этот город был выбран для проведения семинара не случайно. Воронежцы не только активно осваивают вычислительную технику, интенсивно применяя ее в учебном процессе, но и производят компьютеры.

Четыре дня шла работа совещания. Пленарные заседания, где специалисты обменивались опытом, перемежались с практическими занятиями.

В работе совещания принял участие заместитель начальника Главного управления Минвуза РСФСР П. Ф. Анисимов. На пленарном заседании в первый день работы совещания он выступил с докладом, в котором была проанализирована нынешняя ситуация с оснащением и использованием ВТ в средних специальных учебных заведениях республики. Лекцию прочитал Я. Е. Львович, заведующий кафедрой САПР Воронежского политехнического института. О формировании системы непрерывной подготовки специалистов по вопросам информатики и вычислительной техники рассказал председатель методической комиссии Минвуза СССР директор Рязанского техникума электронных приборов И. Орешков.

Выступление Ю. Молоткова, председателя методического объединения, было посвящено работе научно-методического совета Новосибирска по компьютеризации средних специальных учебных заведений. Опыт Новосибирска, где, претворяя в жизнь постановления партии и правительства по внедрению ВТ, действуют совместно с пар-

тийными органами, воспринят с интересом. Партийные руководители города и области регулярно обсуждают ход выполнения соответствующих постановлений и оказывают действенную помощь техникумам.

Научный редактор журнала «Информатика и образование» Т. Драгныш остановилась на том, что мешает журналу стать полноценным помощником преподавателям информатики средних специальных учебных заведений. Крайне маленький штат редакторов; большой разброс в требованиях трех читательских групп: школьных учителей, преподавателей техникумов и профтехучилищ; невысокие полиграфические возможности и т. д.— вот те моменты, которые не дают пока выйти на уровень, о котором мечтает коллектив редакции. Было отмечено, что журнал тем не менее читатели «приняли», тираж его растет. В итоговый документ совещания был записан пункт о необходимости развивать журнал.

Об использовании прикладных программ при изучении курса ОИВТ рассказал Л. Пузыня, преподаватель Ленинградского мореходного училища. Ю. Емельянов, преподаватель Новосибирского техникума электронных приборов, рассказал о системе «Барьер».

Беседы, дискуссии продолжались и в перерывах. Программа семинара была насыщенной. Участникам была предоставлена возможность побывать в авиационном и в электромеханическом техникуме железнодорожного транспорта и в лабораториях Воронежского политехнического института, где можно было записать ППС. Уезжая, многие (особенно представители г. Хабаровска, Магадана) просили организаторов проводить такие совещания-семинары как можно чаще.

## Дети, компьютеры и плодотворные идеи

При традиционных формах школьного образования проблема мотивации учения разрешалась довольно просто — обязательностью школьного обучения для детей, достигших определенного возраста. По мере развития социальных отношений и средств производства настоятельнее ощущается потребность в сознательном и творческом восприятии гражданами стоящих перед обществом проблем. В этом случае одной обязательности обучения как средства формирования мотивации учения уже недостаточно. Необходимо учитывать факторы, влияющие на формирование интереса к учению, на возникновение у ребенка желания познать окружающий мир.

Как свидетельствует психологическая наука, существенным фактором всякой мотивации, а значит, и мотивации учения является *цель*, которую ставит перед собой сам человек (а не только общество перед ним). В процессе целеполагания, т. е. постановки цели, наиболее значительным моментом является проблема *смысла*. Все, что для данного человека лишено смысла, никогда не станет его *собственной* целью. Например, он будет крайне формально относиться к своим обязанностям, даже очень важным с точки зрения внешнего наблюдателя, если те никак не соотносятся с его внутренними целями, т. е. лишены для него смысла.

Итак, если мы хотим формировать сознательное и творческое отношение к учению, нам следует подумать над тем, *каким образом* учение может стать для наших детей целью, наполниться для них смыслом. И в этом случае нам опять-таки придется обратиться к психологической трактовке условий осмысления (придания смысла) индивидом тех или иных событий. Важнейшим из них является *эмоциональная окрашенность* этих событий, т. е. вовлечение в эти события аффективной сферы человека.

Формирование интереса к учению — одна из тех проблем, которым посвящена книга американского математика и психолога профессора Сеймура Пейперта «Переворот в сознании: Дети, компьютеры и плодотворные идеи»\*. Если взглянуть на формирование интереса к учению в условиях современной

школы, то ее позитивное решение кажется маловероятным. Учитель работает с 30—40 детьми, прибегая к дисциплинарным воздействиям, а не решая проблему, как заинтересовать каждого из обучаемых. Предоставить ученикам право выбора того, *чему* обучать, он также не может, поскольку учебные предметы должны усваиваться по специальным программам. Неудивительно, что увлеченность учением детей начальных классов сменяется формальным и равнодушным (а иногда и негативным) отношением к учебе у старшеклассников. Повидимому, одним из серьезных препятствий формирования интереса к учению являются существующие *способы организации обучения в школе*.

Эти способы препятствуют выявлению связи формальных школьных знаний с конкретным жизненным опытом ребенка или, говоря словами С. Пейперта, в современном арсенале педагогических средств отсутствует «переходный объект», устанавливающий связь между абстрактными и чувственными знаниями. Поскольку таким «переходным объектом» для каждого человека является особый предмет, то обобщить сведения в «переходном объекте», превратив это понятие в средство обучения не представляется возможным. Почему «переходный объект» — всегда особый предмет, видно из предисловия, в котором С. Пейперт рассказывает, какую роль в формировании его интереса к учению сыграли передаточные механизмы. Вот что он пишет по этому поводу: «Я полюбил передаточные механизмы! А это нечто такое, что невозможно описать в чисто «когнитивных» терминах. Со мной произошло что-то очень личное, и нельзя утверждать, что у других детей оно повторится в точно такой же форме».

В чем видит выход сам С. Пейперт? Он довольно убедительно разрабатывает идеи Пиаже применительно к развитию аффективной сферы ребенка. Залог успехов в этой области — создание особой среды обучения, которую можно обеспечить, лишь используя компьютер. «Компьютер — это технический Протей. Его сущность в универсальности, в способности к имитации. Поскольку он может принимать тысячи ликов и выполнять тысячи функций, он может удовлетворить тысячи вкусов». Такова точка зрения автора представляемой книги, и задачу свою он видит в том, чтобы

\* *Papert S. Mindstorms. Children, Computers, and Powerful Ideas. Basic Book, Inc., 1980.*

«превратить компьютер в инструмент настолько гибкий, что многие дети сумели бы создать для себя нечто похожее на передаточные механизмы моего детства».

В книге С. Пейперта читатель найдет не только интересную программу использования компьютера для развития способов обучения, но и оригинальную *теорию обучения*. В центре ее — соотношение структур интеллекта ребенка, которые мы хотим развить, и среды обучения, способствующей или препятствующей развитию этих структур. Наиболее благоприятной средой являются создаваемые самим ребенком микромиры. Это микромиры достаточно просты и в тоже время вполне конкретны, как, скажем, мир Черепашки, которая помогает детям знакомиться с геометрическими понятиями.

Разумеется, ребенок, создавая свои микромиры, сохраняет интерес к учению не только потому, что взрослые предоставили в его распоряжение компьютер. Необходимо и преобразование всего учебного процесса. В одной из глав книги описывается, каким может быть этот процесс. Характерной его особенностью является такое обучение, которое строится на *общении* более опытного и знающего взрослого и менее опытных и знающих учеников, а не только на передаче первым знаний. Такое общение продуктивно, если взрослого и детей связывают *общие цели*. Компьютер и играет

роль средства, объединяющего эти цели; его применение делает ученика значимым и равноправным участником ситуации обучения.

Взаимосвязь психологических, педагогических наук, конкретных знаний, технологии обучения, с одной стороны, и социальных условий обучения — с другой, в книге С. Пейперта прослеживается достаточно подробно, хотя педагогические концепции американского ученого, несомненно, будут не только осмысляться, но и дискутироваться.

В заключение отметим привлекательную особенность книги — ее язык. Он отвечает теоретическим установкам автора о важности эмоциональной сферы. Яркость и образность изложения превращает чтение книги в увлекательное занятие. Надеемся, что при переводе на русский язык книга не утратит стилистического обаяния оригинала.

Книга адресована специалистам в области педагогической психологии и разработчикам обучающих программ. Она будет интересна и учителю.

А. ФЕДИНА

От редакции. Выпуск в свет книги С. Пейперта «Переворот в сознании: Дети, компьютеры и плодотворные идеи» запланирован издательством «Педагогика» на 1989 г.

## Скупой платит дважды

На Западе многие опасаются роботов. В фантазии о том, что они могут взбунтоваться, захватить власть, поработить людей, никто сейчас не верит, а вот потеря работы с приходом на завод новой техники — опасность вполне реальная. Однако и этот аспект придется, видимо, рассматривать по-новому. Вот яркий пример.

Плохие условия труда в красильном цехе — жара, пыль, загазованность — вызвали протесты работников, однако хозяин предприятия не хотел вкладывать средства в улучшение вентиляции. Он предпочел поставить на этот



участок роботов. Однако те с первых дней стали ломаться. Вскоре выяснилось, что причина этого... жара и пыль.

## Карточка-компьютер

Крупнейшая международная финансовая сеть «Виза интернэшнл» обслуживает 140 млн. клиентов. Инженеры японского электротехнического концерна «Тошиба» разработали для нее кредитную карточку с замечательными свой-

ствами. Предварительные операции производятся с помощью встроенной ЭВМ, снабженной батареейкой электропитания, буквенно-цифровой клавиатурой и плоским видеозэкраном на жидких кристаллах. Размеры карточек не больше обычных. По сути, это электронная записная и чековая книжка плюс часы плюс справочник по обменному курсу валют. А как потратить деньги при помощи этой системы? Клиент набирает свой код, сумму, которую он хочет снять. На экране воспроизводится остаток вклада, а далее используются терминалы ЭВМ в банках, торговых предприятиях, в сервисных организациях — словом там, куда протянулись линии компьютерной связи с банками а это значит — почти везде...



пустение пока не угрожает. Микросхемы из нового материала еще долго будут сосуществовать с традиционными.

## Транзистор без проводника

Первый транзистор был биполярный — массивный (граммы!) кусок полупроводника с глубоко проникающими областями эмиттера и коллектора. Несколько позже появился полевой транзистор — фактически рисунок на поверхности кремниевого кристалла толщиной в считанные микроны. Однако и он содержал в себе полупроводник. Полупроводник и транзистор были неразлучны — германий, кремний, арсенид галлия, алмаз...

Транзистор без полупроводника? Нонсенс! И тем не менее старой дружбе приходит конец. Самая последняя новость транзисторной технологии — прибор из чередующихся слоев металл-диэлектрик с толщинами в несколько нанометров ( $1 \text{ нм} = 10^{-9} \text{ м}$ , такой же порядок имеют размеры атомов). В таких структурах проявляют себя квантовомеханические эффекты, в частности туннелирование электронов, что и позволяет создавать переключающие приборы, по традиции называемые транзисторами. Нанотехнология — так называются приемы, позволяющие создавать подобные структуры, — обещает создание интегральных схем с плотностями порядка  $10^{15}$  элементов/см<sup>3</sup> и быстротой до  $10^{12}$  операций/с.

## Алмазная электроника

Арсенид-галлиевые транзисторы дороже кремниевых. Ведутся исследования по изготовлению транзисторов из... алмаза. Какое еще более дорогое вещество будет использовано для этих целей в будущем?!

Не думайте, что выбор такой экзотической основы продиктован стремлением к повышению себестоимости и, соответственно, продажной цены прибора, как это порой случается. Специалисты японской фирмы «Самитомо электрик индастриз» пытаются использовать высокую теплопроводность алмаза для более эффективного отвода выделяющегося при работе микросхем тепла, что позволит увеличить их быстродействие. Кроме того, значительно большая, чем у кремния, ширина так называемой запрещенной зоны делает изготовленные из него электронные приборы значительно более устойчивыми к воздействию радиации, а это весьма важно для их применения в космических исследованиях, атомной энергетике, в военной и других областях.

## Прощай, Кремниевая долина?

Про арсенид галлия многие уже слышали: сейчас этот материал считается наиболее перспективным для изготовления полупроводниковых приборов. Он намного дороже кремния, но имеет перед ним существенные преимущества.

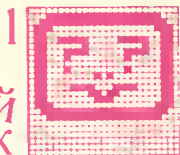
Первое заключается в специфике атомного строения арсенида галлия — движущийся в его кристалле электрон ведет себя так, как будто он имеет меньшую, чем на самом деле, массу. В результате электрическое поле той же напряженности, что и в кристалле кремния, разгоняет электрон до более высокой скорости, а это повышает быстродействие арсенид-галлиевых транзисторов. Ну а быстродействие — один из ключевых факторов микроэлектроники.

Второе преимущество — большая длина свободного пробега электронов (среднее расстояние, проходимое электроном от одного соударения с атомом полупроводника до другого). Естественно, средняя скорость электрона с учетом потерь на соударения намного меньше, чем при «свободном полете». И вот недавно удалось создать так называемые транзисторы на горячих электронах, в которых толщина слоя арсенида галлия между электродами равна лишь  $0,035 \text{ мкм}$ , что меньше свободного пробега в этом полупроводнике. Одержана еще одна победа в борьбе за быстродействие!

Однако Кремниевой долине — песчаной пустыне на востоке США, где расположена большая часть американских заводов по выпуску полупроводников, — за-



Первый советский серийный транзистор П-1 уже не генерирует, не усиливает и не переключает. Максимально огражденный от внешних воздействий, он сохраняется для потомков.



### ДОСКА ОБЪЯВЛЕНИЙ

- Училище готовит операторов ЭВМ и машинного доения...
- Срочно куплю полное собрание программ для ЭВМ...
- Готовлю программистов-стенографистов по сокращенной программе...
- Сдаю программы ЭВМ только с первого предъявления!..
- Пропал исходный модуль! Нашедшего просим вернуть отлаженным за приличное вознаграждение...
- Помните: 60 кг нераспечатанных листингов сберегают одно дерево!
- Читаю и перевожу со словарем с Фортрана, Алгола, Кобола и др.
- Вывожу новые породы ЭВМ с заранее заданными характеристиками...

### ВНИМАНИЕ!

- Создав систему, посмотри — не операционная ли она!
  - Будь объективен к объектному модулю!
  - Обучая начальника работе на персональной ЭВМ требуй персональной надбавки!
  - Категорически запрещается сушить супервизор на интерфейсе!
  - Помни, что ЭВМ женского рода.
- Г. ЮДИН**

### ВОПРОСЫ, ВОПРОСЫ, ВОПРОСЫ...

- А все-таки, стоит ли брать банк данных?
- И почему нет доплат за знание алгоритмических языков?
- Стоит ли писать программу, если заранее знаешь, с какого оператора она начнется и каким закончится?
- Надо ли материально поощрять специалиста, если ему и так дана возможность попрограммировать?

**Е. КОРЖЕНЕВСКИЙ**

● Болеющие душой за дело внедрения в школу компьютеров, скорее выздоровейте: вы очень нужны!

● Глобальная учебная цель оправдывает убогие программные средства.

● Где, укажите, педагогики отцы, которых мы должны принять за образцы?

(почти по Грибоедову)

● Почему портфель ученика стал тяжелее, если компьютер призван избавить людей от рутинной работы, облегчить школьнику учение?

● Редактор — это переводчик с авторского языка на человеческий.

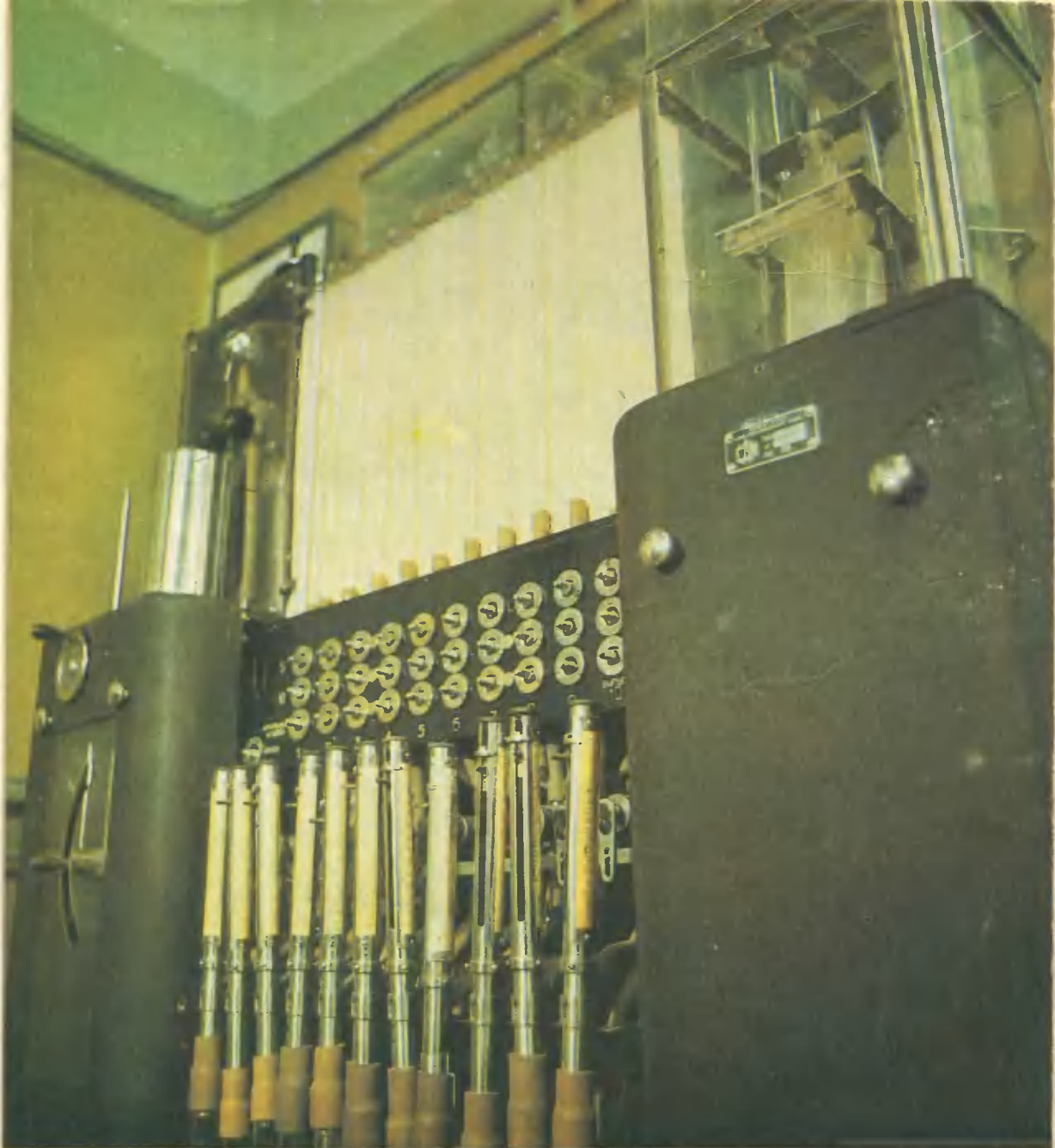
**Т. ДРАГНЫШ**

### ВАРИАЦИИ...

- «Не учите меня программировать!»
- «Программа под редакцией...»
- «Компьютер с девичьей памятью...»
- «Что может сравниться с программой моей...»
- «Полюбите программиста...»

**Ю. ШЕХОНИН**





На фотографии, сделанной в Политехническом музее,— одномерный малогабаритный гидравлический интегратор ИГ-3 конструкции В. С. Лукьянова. В 1935 г. ему было выдано авторское свидетельство на изобретение гидравлической модели «для решения задач по вопросам о нагревании и охлаждении твердых тел и распределения в системах теплового состояния». Подобные устройства играли очень большую роль в «до-электронную» эру, и даже появление ЭВМ не сразу сделало их ненужными. Экземпляр, который вы видите, изготовлен в 1955 г.; почти такой же использовался, например, в Антарктиде во время второй советской экспедиции в 1956—1958 гг. Вычислительную технику в музее можно не только демонстрировать, но и использовать для совершенствования музейного дела (читайте статью «Компьютер в музее»).

На четвертой странице обложки: Гулливер у ЕС-1045! Нет, это обычный ребенок рассматривает уменьшенную копию этой ЭВМ. Созданная в 1979 г., ЕС-1045 занимает площадь 60 м<sup>2</sup>. Машина того же класса, которую будут выпускать в 1999 г., поместится на столе, как и этот макет.

Цена 60 коп.  
70423

073

ИНФО



3'88

ИНФОРМАТИКА  
И ОБРАЗОВАНИЕ

