

ISSN 0234-0453

# ИНФОРМАТИКА И ОБРАЗОВАНИЕ

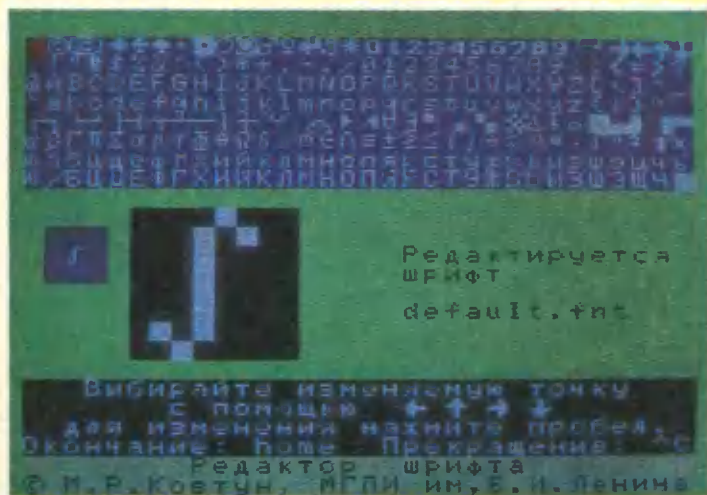
4 1988





## АОС «Радуга»

Кадры справа демонстрируют работу редактора шрифта. Красный квадрат указывает место символа, подлежащего замене, — в данном случае на знак интеграла. С помощью редактора шрифта можно изменить назначение любой клавиши, т. е. задать любой нестандартный символ, необходимый для создания АОК.



• Об автоматизированной системе подготовки обучающих курсов «Радуга» читайте в этом номере



# ИНФОРМАТИКА И ОБРАЗОВАНИЕ

## Содержание

### Общие вопросы

Медведь Б. Учитель и информатика 3

### Методика обучения

Гейн А., Житомирский В., Линецкий Е., Сапир М., Шолохович В. Программно-методический комплекс по курсу школьной информатики 9  
Шакирова Д. Проектирование программно-методического обеспечения для предметов физико-химического цикла 19  
Лебедева М. Анализ содержания учебных предметов для создания педагогических программных средств 22  
Васильевский И. О содержании учебных компьютерных программ 25  
Аншаков О., Писаренко Н. Методические рекомендации по использованию машинной графики БК-0010 26  
Климов Д., Петров М., Урнов В. «Первые шаги» и «Большие проекты» 34  
Копылов Г. Из рукописи конкурсного учебника по информатике 42

### КВТ

Архангельский А. Мир ЭВМ 45  
Штильман Б. РИФ 62  
Буняев М., Давыдов И. Автоматизированная система подготовки обучающихся курсов «Радуга» 65  
Брик И. Простые программы на Фокале 68  
Касък К., Яаксоо К. Решатель задач Solver 70  
Бондаровская В., Коваленко А., Миронченко С., Повякель Н. Автоматизированное рабочее место учащегося: эргономика и гигиена 71  
Доскин В., Козловский С. Психометрическая оценка качества изображения на дисплеях ЭВМ 75  
Збаровский В., Марат Б. Комплексное исследование использования ЭВТ 79  
Бейсик проник еще в одно электронное устройство 82  
Интерпретатор алгоритмического языка 82

### Педагогический опыт

Вейцман В. Старт в информатику 83  
Скобелев Г. Наш эксперимент 87

### Молодежная инициатива

89

## Точка зрения

Гегечкори А. Проблемы обучения информатике	99
Шовман М. Чего не хватает уроку информатики	105
Кушир М. Компьютеризацию начали. Что впереди?	107

## Педагогические кадры

Марков С. Компьютеризированная система непрерывного повышения квалификации работников профтехобразования	109
--	-----

## Внеклассная работа

Стернин М. Тысяча лиц	112
-----------------------	-----

## ЭВМ в народном хозяйстве

Толстошеев В. Автоматизированное рабочее место арбитра	115
«Мертвая зона», или Павильон № 2	118

## Нам пишут

Срочно требуется задачник!	119
«Имитация с целью симуляции»	120
Из опыта эксплуатации комплексов учебной вычислительной техники	121

## Информация

Переподготовка преподавателей по информатике и вычислительной технике	122
Роберт И. Программно-методическое обеспечение школьных ЭВМ	123
Всероссийский семинар-практикум	124
МНР: компьютеризация образования	125
О печальном и забавном	126

Обложка Э. Бажилина

В оформлении номера принимали участие Э. Бажилин, С. Кустарева, 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. Драгныш  
Заведующая редакцией Н. Игнатова  
Художественный редактор Л. Розанова  
Корректоры Е. Власова, М. Суворова

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 24.05.88. Подписано в печать 07.07.88. А 05714  
Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 10,40. Уч.-изд. л. 13,63.  
Усл. кр.-отт. 42,88. Тираж 95 040 экз. Заказ 1327. 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

**Б. МЕДВЕДЬ**

зав. учебно-методическим кабинетом ИВТ Ленинградского городского ИУУ

## Учитель и информатика

### Проблемы профессионального и творческого роста

Одной из наиболее важных проблем становления нового школьного курса является подготовка учителей, включающая такие аспекты, как организация, концепция, перспективы. Рассмотрим их, базируясь на опыте, накопленном в Ленинграде.

#### Организация подготовки учителей

Общая схема курсовой подготовки. На первом этапе (1985/87 уч. г.) подготовка педагогов велась по «географическому» признаку: за 21 районом города было закреплено 20 ведущих вузов, которые силами своих специалистов за два года (1985 и 1986 гг.) на собственной базе подготовили более 600 учителей информатики. Кроме того, ЛГПИ им. А. И. Герцена подготовил резерв руководящих кадров городских школ, работников роно (всего около 300 человек), учителей области (более 200 человек); Ленинградский институт авиационного приборостроения — учителей программирования и информатики для школ с углубленным изучением физико-математических дисциплин (около 200 человек), а Ленинградский институт методов и техники управления — более 70 методистов всех предметов городского ИУУ.

Эта работа проходила в условиях тесной координации между органами народного образования и Советом ректоров вузов. Прикрепление курирующих вузов к районам, осуществленное на первом этапе введения в школы нового предмета, позволило равномерно распределить ресурсы города, решить поставленные задачи в короткий срок, создать районную инфраструктуру школьной информатики — фундамент для дальнейшей консолидации уси-

лий научно-педагогической общественности в районах. **3**

На следующем этапе (подготовка к 1987/88 уч. г.) было решено сгруппировать учителей по производственному признаку (по типу вычислительной техники, на которой они работают с учениками). Сохранить прежнее прикрепление удалось только там, где первоначально соблюдался территориально-производственный принцип.

В настоящее время учителя Ленинграда работают с учащимися на ЕС ЭВМ, СМ ЭВМ, «Искрах», ДВК, КУВТ-86, «Агатах», «Ямахах», ДЗ-28, «Синклерах», «Правцах»; почти каждый из этих типов имеет несколько модификаций. Главная причина такой пестроты в том, что из-за недостаточных централизованных поставок приходится широко привлекать технику базовых предприятий.

Итак, летом 1987 г. было организовано 10 групп учителей (две из них работали с КУВТ-86 разных типов). Второй поток работающих на КУВТ-86 (еще 2 группы тех, кто «не поместился» на практику летом) продолжал обучение в течение учебного года.

Курсовая подготовка велась по учебным планам, разработанным в Ленинграде с учетом местных условий. Она включала 18 ч общих лекций по методике и передовому педагогическому опыту и 54 ч занятий на компьютерах. В практической части особое внимание уделялось наработке опыта ведения занятий, накоплению вариантов задач для учащихся, приемам демонстрации возможностей компьютера в обработке разнообразной информации. В качестве выпускной работы учитель должен был подготовить разработку одного урока информатики по своему выбору.

Состав и статус учителей. Необходимо отметить, что учителя информатики (речь идет прежде всего о тех, кто ведет курс с осени 1985 г.) выдержали серьезные испытания: три лета подряд — переподготовка (суммарный объем — более 300 ч), многочисленные мероприятия в течение учебного года, постоянные хлопоты с кабинетом и оборудованием или с вождением учащихся на чужую базу, многочасовая подготовка к каждому уроку, неизбежный трепет перед вопросами «всезнающих» учеников и т. п. Отметим, что большинство учителей (по крайней мере, на первых порах) преподавали информатику «в нагрузку» к математике, получая весьма скромную материальную компенсацию за свои усилия. При ответственном подходе на подготовку к 2 ч информатики приходилось тратить почти столько же времени, сколько для 20 ч математики, неся при этом и психологическое бремя: не может же, хороший педагог допустить, что он что-то, пусть даже в новой предметной области, знает хуже, чем ему бы хотелось!

4

Далеко не каждый способен выдержать такие трудности, поэтому мы не удерживали административно тех, кто после первого года преподавания информатики решил расстаться с предметом. Напротив, мы благодарили их за выполненный долг в труднейший первый год. Зато об оставшихся можно сказать: в большинстве это люди, преданные делу компьютеризации образования. Хотя для системы повышения квалификации уход уже подготовленных специалистов — процесс болезненный, мы считаем, что добровольный выбор предмета учителем — одно из обязательных условий успешной работы. Преподавание — сложный творческий процесс, а творчество невозможно по принуждению.

Приведем некоторые данные о составе ленинградских учителей информатики по состоянию на начало 1988 г.

Всего средних школ в Ленинграде	337
Всего учителей информатики	363
Из них:	
основных учителей школы	292 (80 %)
совместителей с производства	71 (20 %)
ведут только информатику	101 (28 %)
учителей по диплому	190 (52 %)
инженеров по диплому	151 (42 %)
математиков, физиков, экономистов по диплому	22 (6 %)

Из учителей-профессионалов — подавляющее большинство математиков (93 %),

гораздо меньше физиков (5 %).

В Ленинграде создана и работает схема социалистического соревнования, активно направляющая процесс формирования контингента учителей информатики. В соответствии с ней большее количество баллов получает тот район, где среди учителей информатики:

меньшее (по отношению к основным учителям) число совместителей;

большее число учителей, ведущих курс не первый год.

Подобный подход позволяет активно влиять на политику подбора, расстановки и воспитания кадров на местах. Уменьшается количество совместителей с производства (учитель, работающий в районном центре школьной информатики на несколько школ, совместителем не считается).

Важно отметить, что значительная часть преподавателей (до 75 % к 1988 г.) не собираются отказываться от ведения информатики.

Постепенно нормализуется положение по части материальной компенсации затрат учительского труда: речь идет как об оплате за кабинеты и работающие дисплеи, так и об увеличении доли информатики в общей нагрузке учителя, ведущего и другой предмет.

Необходимо отметить обстановку доверия, в которой работают учителя информатики. Органы образования города нацеливают районные звенья и администрации школ на оценку работы учителя по существу, по конечному результату, а не по застывшей рецептурной схеме («на какой минуте 7-го от начала полугодия урока следовало впервые произнести слово «алгоритм»?»). Естественно, это приветствуется большинством учителей.

Межкурсовая подготовка учителей. Именно работа в межкурсовой период превращает последовательность дискретных мероприятий в систему непрерывного образования. Перечислим основные формы этой работы: групповые и индивидуальные консультации по основным вопросам курса (ЛГИУУ); занятия народного университета по информатике в Ленинградском Доме научно-технической пропаганды (основной контингент — учителя и преподаватели СПТУ и техникумов); семинар по передовому педагогическому опыту (ЛГИУУ); обеспечение учителей методической литературой (ЛГПИ им. А. И. Герцена, ЛГИУУ, НИИ общего образования взрослых АПН СССР); участие учителей в ежегодной научно-технической конференции «Школьная информатика»; участие в организации школьного и районного туров олимпиады по информатике; привлечение школьников к кон-

курсу рефератов (городская научно-техническая конференция старшеклассников); участие в научном семинаре межотраслевого комитета по проблемам образования в области информатики, вычислительной техники и автоматики при Ленинградском областном Совете научно-технических обществ.

Ясно, что идеи компьютеризации образования должны пронизывать всю школу — только в этом случае новый учебный предмет будет жизнеспособным. Поэтому мы уделяем большое внимание «компьютерному ликбезу» для всех учителей-предметников и руководителей народного образования. Он включает в себя: проведение краткосрочных курсов по 36-часовой программе для работников народного образования, не преподающих информатику (ЛГИУУ); разовые занятия для учителей разных дисциплин (ЛГИУУ совместно с ЛО Педобщества РСФСР); выездные педсоветы по тематике «Пути интенсификации учебного процесса» в кабинете ИВТ (ЛГИУУ); привлечение учителей-предметников к созданию проектов сценариев педагогических программных продуктов (ЛГИУУ); проведение проблемных курсов для всех предметников одной школы или энтузиастов одного района.

## Концепция

Прошла «пожарная» спешка первых лет. Основной состав учителей информатики подобран и действует. На повестку дня ставятся новые вопросы.

Что нас не устраивает в подготовке учителей сегодня?

Анализ посещаемых уроков, анкетирование учителей, беседы с ними, размышления о тематике курса дают следующие ответы на этот вопрос.

Значительная часть учителей пока весьма поверхностно представляют себе научно-методологическую основу курса информатики. В лучшем случае они искренне считают, что это — программирование для ЭВМ. Мы же убеждены в том, что необходимо всячески подчеркивать разницу между обучением информатике (фундаментальная дисциплина, аккумулирующая приемы и формы работы с информацией во всевозможных предметных областях человеческой деятельности) и программированию (прикладная технологическая дисциплина, о которой непрофессионалу достаточно иметь лишь общее представление).

Часто на учительских семинарах и конференциях мы становимся свидетелями и участниками бесконечных и бесполезных споров о том, какой язык программирова-

ния лучше, зачем придуман еще один язык, когда уже есть такой хороший (или такой плохой) Бейсик и т. п. На наш взгляд, живучесть интереса к этим вопросам основана на непонимании того фундаментального факта, что информатика в нашей школе должна стать общеобразовательным, а не узкопрофессиональным предметом, и, следовательно, важно изучать алгоритмы, а не языки.

Для каждого учителя (и ученика!) должен стать совершенно очевидным выбор: лучше хорошо описывать алгоритмы на произвольном языке, чем плохо — на заданном! Приходится терпеливо разъяснять неправомерность возвеличения одного языка и принижения возможностей других. Хорошо зарекомендовал себя на уроках метод параллельного показа текстов одной несложной программы, написанных на нескольких языках. Этот прием, по-видимому, имеет серьезную педагогическую ценность, ибо он великолепно срабатывает и на школьников, и на взрослых: удается довести до сознания слушателей важнейшую мысль о том, что между языками значительно больше сходства, чем различий. Наши ученики уже не ответят на провокационный вопрос «Какой язык лучше: Фортран, Рапира или Паскаль?» без предварительного выяснения — для какого класса задач?

Постепенно нам удается воспитывать у учителей более уважительное и терпимое отношение к различным языкам. Это происходит и потому, что на олимпиадах по информатике до районного тура включительно школьнику разрешается применять любые версии любых языков программирования (на городском туре выбор ограничен «всею» девятью языками из-за проблем с трансляцией: Алгол-68, Фортран, Паскаль, Бейсик, Си, Рапира, ПЛ/1, Фокал, алгоритмический язык учебного пособия). Нет худа без добра: выработке широких взглядов на обсуждаемую проблему способствовало и такое негативное явление, как вынужденная работа ленинградских школьников на множестве различных типов ЭВМ.

Многие учителя считают, что информатика — это раздел математики. Им не удалось пока избавиться от рассмотрения «арифметизации» как единственного (в их понимании) способа обработки информации. Они находятся в плену слова «вычислительная» в аббревиатуре «ЭВМ» и искренне полагают, что ЭВМ умеет лишь вычислять.

Дело осложняется тем, что формально они правы. Но мы убеждены, что «внутренняя жизнь» машины не должна превалировать в процессе общеобразовательной подготовки (ведь все пользуются телефоном, но далеко

не все знают устройство квазиэлектронной АТС).

Учителя, о которых идет речь, не видят пока широчайшей межпредметной базы информатики; это как раз та ситуация, про которую сказано: «За деревьями не видят леса». Преподаватели идут напролом, когда надо совершить маневр; в который уже раз терзают они надоевший всем алгоритм решения квадратного уравнения, считая, что приносят пользу ученику и агитируют за математику.

После одного из подобных серых уроков выяснилось, что учительнице математики с 10-летним стажем, имеющей и музыкальное образование, не приходило в голову, что на уроке информатики не только можно, но и весьма желательно дать примеры кодирования нот, а потом продемонстрировать приемы символьно-цифрового представления мелодии в ЭВМ и тем самым исподволь подойти к математике — не по своему умыслу, а по природе вещей, потому что «природа говорит с нами на языке математики».

6

На наш взгляд, именно узкоспециализированное «физико-математическое» преподавание информатики — основная причина потери учащимися интереса к новому предмету, с чем, к сожалению, нередко приходится встречаться. Действующее учебное пособие также пропагандирует одностороннюю информатику, практически не затрагивает ее гуманитарно-лингвистических и изобразительно-звуковых аспектов, не дает представления об истоках и предпосылках этой области, о закономерном характере ее появления и развития, слабо демонстрирует связи ЭВТ с жизнью современного общества. Апробация в нескольких классах ленинградских школ объединенного учебного пособия А. П. Ершова, А. Г. Кушниренко, Г. В. Лебедева, А. Л. Семенова, А. Х. Шеня позволяет надеяться, что оно поможет лучше решать затронутые проблемы (хотя и оно нуждается, на наш взгляд, в модификации).

Определенная часть учителей, имея довольно узкий кругозор, ограниченный лишь школой, и низкую познавательную активность, восприняли введение нового предмета как очередную кампанию, которых нашему образованию, к сожалению, было не занимать. Эти педагоги психологически не готовы к восприятию новой реальности: мир вступает в эпоху информационного общества; на повестку дня выходит важнейшая категория — информационные ресурсы общества; человек вручную более не в состоянии эффективно управлять уплотняющимися и ускоряющимися потоками всевозможной информации, обрабатывать их.

Обозначив основные «претензии» к сегодняшним учителям информатики, попробуем наметить дальнейшие направления работы с ними.

### Перспективные формы работы с учителями в области информатики

Как уже отмечалось, одна из важнейших задач работы с учителем — привитие ему интереса к приобретению знаний, передача убежденности в правильности пути, по которому он ведет учеников, расширение кругозора. На наш взгляд, во всех формах повышения квалификации учителя заметную роль должны играть приемы мировоззренческого плана. Побольше творческих заданий, лекций и семинаров о глобальных проблемах, возникающих перед человечеством, о реальностях, трудностях и неотвратимости информатизации общества, о механизмах, сдерживающих эти процессы, о достижениях и неудачах!

Следует, например, приветствовать издание в НИИ ООБ АПН СССР учебно-тематических планов и программ проблемных семинаров по философско-методологическим и историческим аспектам информатики. Используя эту тематику, широко охватывающую проблемы современного общества, мы должны помочь учителю не пичкать ученика знаниями по готовой рецептуре, а сотрудничать с ним в обсуждении и поисках решения трудных вопросов. По нашему мнению, информатика, как никакой другой сегодняшний школьный предмет, подтверждает педагогику сотрудничества (впрочем, мы убеждены: без сотрудничества в решении проблем нет никакой педагогики!), педагогику проблемную взамен «педагогики» нормативной.

На вводных занятиях по информатике в ЛГИУУ приходится видеть, как учителя (в первую очередь других предметов, надеявшиеся, придя на непрофильную лекцию, тихо отсидеться) начинают оживать, внимательно слушают, задают вопросы, приводят собственные примеры, вступают в дискуссию.

Конечно, встречаются и недовольные. Некоторые учителя, привыкшие не обсуждать пути поиска, а диктовать, и сами ожидают только диктовки. Они лучше принимают простую лекцию о назначении клавиш микрокалькулятора (ее, уже «разжеванную», можно целиком вложить в ученика), чем взволнованный рассказ об ошибках и инвестиционной политике, приведших к тяжелому сегодняшнему положению с вычислительной техникой. Творчески работающий учитель любит выбирать порядок и формы прохождения материала сам, а привыкший работать по застывшей схеме предпочитает, чтобы все



выбрали за него. Именно учителя второго типа ушли от преподавания информатики в последние год-два, что в конечном счете следует приветствовать.

По существу, мы лишь подтверждаем выбор, который сделан партией и народом: больше демократии! Демократизация в педагогике проявляется и в положении учителя, и в отношении к нему инспектора, и в отношении учителя к ученику, к программе курса.

Нам уже удается привлекать учителей к участию в написании проектов сценариев обучающих программ. Сами программы под руководством учителя пишут школьники либо студенты.

Например, во время занятий по информатике с учителями биологии (несколько лекционных и практических часов) к нам обратился учитель СШ 248 Кировского района В. А. Смирнов. Проявив интерес к использованию компьютера в обучении биологии, он получил возможность дополнительно посещать кабинет ИВТ после окончания курсов. После нескольких рабочих посещений, в течение которых он более подробно ознакомился с возможностями КУВТ «Ямаха» и имеющимися обучающими программами (в том числе и по другим предметам), у него зародилась идея построения программы, которая поможет шестиклассникам в выработке навыка узнавания вида растений (работа с определителем растений). Отметим, что подобная работа (фактически — постановка диагноза) — весьма важный элемент и других видов человеческой деятельности.

В помощь В. А. Смирнову мы направили добровольца — десятиклассника физико-математической СШ 30 Василеостровского района В. Павловского, который к тому времени проявил себя сильным программистом. В результате совместной работы сценариста-учителя и программиста-школьника был создан первый вариант программы «Ботаника». Программа демонстрировалась учителям биологии, методистам, прошла апробацию на учащихся. Уроки, проведенные в кабинете ИВТ ЛГИУУ с участием научных сотрудников лаборатории естественно-математических дисциплин НИИ ООБ АПН СССР, записаны на видеомagneтофон, демонстрировались в записи на научно-практической конференции «Перестройка школы: поиски новых форм научно-педагогической деятельности» (январь 1988 г., Ленинград, НИИ ООБ АПН СССР), где выступал и сам В. А. Смирнов.

Сейчас В. А. Смирнов продолжает работу с В. Павловским (ныне студентом) над совершенствованием программы «Ботаника», а с другим учеником (уже из своей школы) —

над созданием программы по цитологии.

В этом примере речь идет, по существу, о возникновении на базе кабинета ИВТ ИУУ центра компьютерного творчества учителей. На наш взгляд, это весьма перспективная форма привлечения учителей к поиску путей повышения эффективности учебного-воспитательного процесса.

Кратко осветим ряд организационных вопросов этой деятельности.

Роль кабинета ИВТ ИУУ: он предоставляет свою материальную базу (КУВТ «Ямаха»), осуществляет координацию работы и организацию содружества «сценарист — программист», использует разрабатываемые программные педагогические продукты (ППП) для учебного процесса и повышения квалификации учителей.

Время работы: вторая половина дня, когда кабинет свободен от курсовых занятий, а авторы программ — от уроков.

Разработчики сценариев: учителя-предметники в содружестве с методистами соответствующих кабинетов ИУУ, научными сотрудниками, преподавателями вузов. Наш кабинет, например, поддерживает контакты с многими кафедрами ЛГПИ им. А. И. Герцена, кафедрой вычислительной техники Ленинградского механического института им. Д. Ф. Устинова. Официальная сторона контактов — договоры о творческом содружестве с этими вузами.

Программисты: студенты указанных вузов, а также учащиеся физико-математических (в основном) школ, которые получают задания, выбирая предметную область из предлагаемых нами. Со школами кабинет ИВТ также заключил договоры о творческом содружестве. Для учащихся это возможность получать задания по производительному труду в соответствии с профилем их трудового обучения и практической квалификации.

Отметим два важных, на наш взгляд, аспекта. Во-первых, взаимоотношения учителя и учащихся, совместно работающих над программой, строятся на принципах, которые в столь концентрированном проявлении не часто встречались в практике советской школы: налицо полное взаимопонимание, равноправие, интерес к общим проблемам — иными словами, подлинная педагогика сотрудничества. Думается, что работа в такой обстановке важна для воспитания, развития, профориентации ученика ничуть не менее, чем сам факт получения результата.

Во-вторых, инициатива снизу, безусловно, должна подкрепляться и встречными мерами сверху. Речь идет о заказе на учебные программы и о мерах поощрения учителей-разработчиков.

Естественно, возникает вопрос: почему бы

силами научно-педагогической общественности с учетом предложений заинтересованных сторон (родителей, предприятий) не составить списки тем, перспективных с точки зрения разработки ППП? Безусловно, такое мероприятие потребует серьезных усилий: надо будет провести системный анализ каждого школьного курса, четко разделить в нем главное и второстепенное, определить, для каких именно разделов привлечение ЭВМ действительно обоснованно, даст наибольший эффект, — но ведь и польза может быть колоссальной! Не придется тогда энтузиастам расходовать силы зря: придумывать или искать вслепую темы ППП, а затем узнавать (хорошо, если скоро!), что произошло дублирование, а другой раздел предмета так и остался белым пятном.

Нельзя, конечно, полагаться в этой работе только на энтузиазм учителя: надо помочь

ему. Конечно, в случае с В. А. Смирновым заведующие кабинетами биологии и информатики ЛГИУУ смогли договориться, чтобы учителю-биологу упомянутая разработка сценария была зачтена в качестве курсовой работы. А может быть, стоит поставить вопрос шире? Например, какая форма повышения квалификации учителя в принципе предпочтительнее: вежливое прослушивание лекций, посещение семинаров и т. п. или мучительная и беспокойная совместная с учащимися работа в компьютерном классе над созданием обучающей программы, которая и после двадцатого пробного запуска таит в себе ошибки, стимулирует своих создателей на новые импульсы творчества?

Очень хочется узнать мнение учителей и организаторов народного образования по этому вопросу.

## Видео АОС

На выставке «Информатика в США» (о ней было рассказано в шестом номере журнала за 1987 г.) демонстрировался «диалоговый аттракцион» — монитор, с экрана которого диктор задавал вопрос; посетитель выставки отвечал, диктор спрашивал что-то еще... Создавалось впечатление беседы с живым человеком.

Основа этого устройства — лазерный видеопроектор, считывающий информацию с диска. На диске диаметром около 30 см записывается состоящая из множества коротких фрагментов-вопросов «телепрограмма» общей длительностью 1—2 часа. В зависимости от очередного ответа человека управляющий системой компьютер выбирает нужную видеозапись, «ведя» диалог.

Обучающие компьютерные программы, использующие аналогичную технику, создает американская фирма «Интерактив Медикал Комьюникейшнз» (ИМК). В этих программах основная задача компьютера — управлять выдачей

# ЧТО? МОЖЕТ ЭВМ

на монитор нужных изображений (в том числе и текстов) в зависимости от реакции обучающегося. Собственно, и в обычных автоматизированных обучающих системах (АОС) главная задача компьютера именно такова, однако использование видеодиска переводит процесс обучения на качественно новый уровень. Причина этого как в большей наглядности видеoinформации, так и в большей гибкости программ, достигаемой благодаря повышению информационной емкости видеодиска по сравнению с магнитным.

Видеосистемы обучают втрое быстрее. Президент ИМК считает, что «диалоговый метод обучения с применением лазерных дисков сейчас единственный практически пригодный метод, гарантирующий, что обучаемые понимают и запоминают важную информацию, которую они смогут затем использовать в работе».

## ЭВМ в артиллерии

Несмотря на все более широкие разработки и внедрение высокоточного управляемого и самонаводящегося оружия, артиллерия пока не намерена уступать ему поле боя. Однако реагировать на веления времени ей приходится. Быстротечность изменения обстановки на поле боя, резкое увеличение числа подвижных целей требуют использования ЭВМ для управления огнем.

В рамках одного или нескольких артиллерийских соединений огнем должны управлять автоматические системы управления, отдельные же подразделения оснащаются портативными ЭВМ.

Компьютеры более ранних разработок хранят в памяти таблицы данных о стрельбе конкретной системы артиллерийского оружия и поправочные коэффициенты для учета конкретных условий в момент стрельбы. Более поздние образцы реализуют математические модели полета снаряда, решая систему дифференциальных уравнений.

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

## Программно-методический комплекс по курсу школьной информатики

Пятая глава посвящена еще одному способу организации действий — выделению вспомогательных алгоритмов. В отличие от ветвлений и циклов выделение вспомогательных алгоритмов не является необходимой формой организации действий. Мотивация для введения этой формы состоит в следующем. Во-первых, выделение вспомогательного алгоритма позволяет сократить и сделать более читаемым основной алгоритм. Во-вторых, вспомогательные алгоритмы можно рассматривать как некие стандартные «детали», которые можно использовать в различных алгоритмах. В-третьих, составляя для исполнителя вспомогательный алгоритм, мы расширяем набор его допустимых действий, как бы обучаем исполнителя новому действию. В-четвертых, вспомогательные алгоритмы являются ступенями в пошаговой детализации алгоритмов. Каждое из этих положений четко формулируется и демонстрируется на ряде примеров.

Важным понятием, возникающим в связи со вспомогательными алгоритмами, является понятие о локальных и глобальных переменных. Вот как оно раскрывается.

Учащиеся объясняют, что при использовании вспомогательного алгоритма, как правило, никого не интересует, из каких действий он состоит. Важно только, каковы его исходные данные (аргументы) и что является результатом его работы. Точно так же, для того чтобы пользоваться автоматом «Газированная вода», не обязательно вникать в его устройство. Достаточно знать, что «аргументом» для него является монета, а «результатом» — стакан воды (с сиро-

пом или без) либо пустой стакан, если автомат неисправен.

Здесь необходимо упомянуть об одном очень важном обстоятельстве. Может случиться так, что в основном алгоритме будут использоваться те же обозначения, что и во вспомогательном. Понятно, что такие переменные надо как-то различать. Поэтому договариваются считать, что все обозначения из вспомогательного алгоритма действительны только в его пределах. Сказанное, конечно же, не относится к результатам работы вспомогательного алгоритма, потому что они должны использоваться и после завершения его работы. Образно говоря, переменные, имеющие одно и то же название в основном и вспомогательном алгоритмах, — просто «однофамильцы». При вызове вспомогательного алгоритма значения переменных основного алгоритма как бы «замораживаются», а после окончания работы вспомогательного алгоритма они «размораживаются».

Далее на конкретных примерах показывается, что произошло бы без такого «замораживания».

Отдельный параграф пятой главы посвящен методу пошаговой детализации алгоритмов. Он объясняется на примере задачи составления для Чертежника алгоритма рисования слова РОБОТ (рис. 9а) (высота каждой буквы — 4 см, ширина — 1 см, расстояние между буквами — 1 см).

Попытка записать алгоритм с ходу вызывает возникающие на этом пути трудности: он получается очень длинным, нерациональным, к тому же наверняка будет содержать ошибки, которые трудно обнаружить. Поэтому учащимся предлагается другой подход к построению алгоритма. Ясно, что Чертежник должен последовательно нарисовать



9

буквы Р, О, Б, О, Т. Таким образом, задача разбивается на пять этапов (рис. 10).

Считая, что вспомогательные алгоритмы рисования букв уже составлены, можно сразу записать основной алгоритм:

10

Выполнить алгоритм «Р».

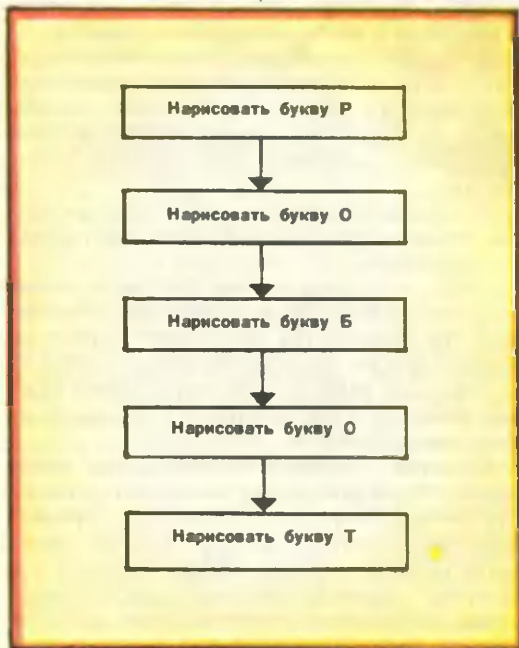
Выполнить алгоритм «О».

Выполнить алгоритм «Б».

Выполнить алгоритм «О».

Выполнить алгоритм «Т».

Теперь можно приступить к составлению вспомогательных алгоритмов. При этом нужно позаботиться о двух вещах: чтобы Чертежник правильно нарисовал отдельные буквы и чтобы, выполнив основной алгоритм, он сложил из них слово РОБОТ. Иначе говоря, надо заботиться не только о форме букв, но и об их стыковке в слове. Это напоминает строительство дома: надо заботиться не только о качестве кирпичей, но и о том,



10

чтобы между ними не оставалось щелей. Таким образом, основной алгоритм предъявляет дополнительные требования к стыковке вспомогательных.

В нашем случае эти требования могут быть, например, такими. Будем считать, что Чертежник начинает рисование каждой буквы с ее левой верхней точки, глядя вправо. Каждый из вспомогательных алгоритмов должен выводить Чертежника в исходную позицию для рисования следующей буквы.

Записав основной алгоритм и определив требования к вспомогательным, мы спустились на более низкий уровень в решении задачи, упростили себе дальнейшую работу. Однако алгоритмы рисования отдельных букв тоже не так уж просты. Значит, составление каждого из них также целесообразно разбить на несколько этапов, спустившись еще ниже. И так далее, до тех пор, пока задачи очередного уровня не окажутся совсем простыми.

Вот характерные задачи, используемые для закрепления знаний по вспомогательным алгоритмам.

1. Для рисования квадрата на рис. 9в был составлен следующий алгоритм:

Выполнить алгоритм «Уголок».

Выполнить алгоритм «Уголок».

Сформулируйте требования к вспомогательному алгоритму «Уголок» и запишите его.

2. Используя метод последовательной детализации, выделите вспомогательные алгоритмы, необходимые для того, чтобы Чертежник повторил рис. 9б. Составьте соответствующий алгоритм.

3. Чертежник помещен на квадратный лист бумаги, сторона которого длиннее 3 см. Составьте для Чертежника алгоритм рисования квадрата со стороной 1 см.

4. Чертежник помещен на прямоугольный лист бумаги. Составьте для Чертежника алгоритм рисования квадрата наибольшей возможной площади. (При решении этой задачи учащимся предлагается воспользоваться ранее решенными задачами.)

5. Составьте для Вычислителя алгоритм, с помощью которого он из трех данных чисел выберет то, которое находится между двумя другими.

Метод пошаговой детализации позволяет продемонстрировать учащимся коллективное составление алгоритмов. На лабораторной работе 8 они разделяются на несколько групп, каждая из которых коллективно выполняет задание (составляет алгоритм рисования некоторого слова). Подпрограмму рисования одной буквы пишет и отлаживает один человек. В каждой группе выбирается руководитель, который создает основную программу рисования слова, определяет требо-

вания к подпрограммам и присоединяет подпрограммы к основной программе.

В главах 2—5 говорится об организации действий, составляющих алгоритм, и совсем почти не затрагивается организация данных, над которыми эти действия производятся. Задачи из этих глав не требуют уделять большое внимание организации данных (используется лишь один способ организации — переменная). В главе 6 учащимся демонстрируется, что решение многих задач было бы просто невозможно, если бы данные не были каким-либо образом организованы: упорядочены, классифицированы, пронумерованы и т. д.

В качестве одного из основных способов организации данных выступают линейные и прямоугольные таблицы. Если переменные систематически используются в курсе математики и в других школьных предметах, то таблицы применяются главным образом как наглядное средство. Поэтому мы включили в главу 6 параграф «Табличный способ организации данных», в котором вводятся понятия таблицы, элемента таблицы, а также способы задания и обозначения таблиц и их элементов. Приведем несколько задач к этому параграфу.

1. Определить, как выглядит таблица А, состоящая из 4 строк и 5 столбцов, если:

а)  $A(i, j) = ij$ ;

б)  $A(i, j) = i$ ;

в)  $A(i, j) = 1$ ;

г)  $A(i, j) = \max(i, j)$

д)  $A(i, j) = \min(i, j)$

е)  $A(i, j) = |i - j|$

ж)  $A(i, j)$  равняется 1, если  $i \leq j$ , и 0, если  $i > j$ .

2. Дана таблица В. Запишите, как выражаются ее элементы  $B(i, j)$  через  $i$  и  $j$ , если таблица имеет вид:

а)

$$\begin{bmatrix} 2 & 4 & \dots & 2^n \\ 2 & 4 & \dots & 2^n \\ 2 & 4 & \dots & 2^n \\ 2 & 4 & \dots & 2^n \end{bmatrix}$$

(в этой таблице 4 строки и  $n$  столбцов);

б)

$$n \ n-1 \ \dots \ 2 \ 1$$

(в этой линейной таблице  $n$  столбцов).

3. В таблице приведены итоги шахматного турнира.

0	1/2	1	0	1	1/2	0	0
1/2	0	1/2	0	1	0	0	1
0	1/2	0	1	0	1	1	0
1	1	0	0	1/2	1	0	1
0	0	1	1/2	0	1/2	1/2	1/2
1/2	1	0	0	1/2	0	1	1
1	1	0	1	1/2	0	0	0
1	0	1	0	1/2	0	0	0

Сколько очков набрал победитель турнира?

В этой главе появляется Робот-манипулятор. Если рассматривать организацию данных так, как это принято в современном программировании, т. е. как единство структуры и правил оперирования, то, как явствует из приведенного выше списка допустимых действий Робота-манипулятора, способ организации его данных отличается от табличного. Можно назвать этот способ организации данных индексно-последовательным.

В этой же главе набор допустимых действий Вычислителя расширяется за счет действий с числовыми таблицами и их элементами (индексированными переменными). При работе с таблицами, естественно, возникает потребность в новой разновидности циклов — цикле со счетчиком. Он оформляется так:

Для каждого I от L до R:

P1

P2

...

Конец цикла по I.

Здесь I — «счетчик», L — нижняя, а R — верхняя границы изменения I; P1, P2... — действия, которые выполняются в цикле.

Выполняя этот цикл, Вычислитель сначала присвоит I значение L. Затем он проверит, больше ли I, чем R; если да, то цикл заканчивается (не начавшись), если нет, то Вычислитель выполнит действия P1, P2, ..., число I увеличивается на 1, и для этого нового значения I повторяется то же самое до тех пор, пока I не превзойдет R.

В качестве примера разбирается задача обработки результатов метеорологических наблюдений. Исходные данные (месячные количества осадков, выпавших в Свердловске в течение 20 лет) расположены в виде таблицы из 20 строк и 12 столбцов.

Приведем некоторые задачи на составление алгоритмов работы с таблицами.

1. Составьте для Робота-манипулятора алгоритмы, выполнив которые он:

а) соберет нужные детали из первого справа столбца стеллажа;

б) соберет нужные детали из третьего сверху ряда стеллажа.

2. Дана таблица А:

3	-2	5	7
6	0	-2	0
8	1	2	0

Какой вид она примет после выполнения следующего алгоритма:

Присвоить  $A(1,1)$  значение 2.

Присвоить  $A(3,2)$  значение  $A(1,1) + A(3,2)$ .

Если  $A(3, 2) > 3$ , то:

Присвоить  $A(3, 4)$  значение  $2A(3, 1)$ .  
Иначе:

Присвоить  $A(1,4)$  значение  $A(3,2) + 3$ .  
Конец ветвления.

Для каждого  $I$  от 1 до 3:

Присвоить  $A(I, J)$  значение  $I^2$ .

Конец цикла по  $I$ .

3. В алгоритме вычисления суммы кубов элементов таблицы  $A$ , имеющей 4 строки и 3 столбца, злоумышленник поменял местами две строки. Вот что у него получилось:

Присвоить  $S$  значение 0.

Для каждого  $I$  от 1 до 4:

Для каждого  $J$  от 1 до 3:

Присвоить  $S$  значение  $S + A(I, J)^3$ .

Конец цикла по  $I$ .

Конец цикла по  $J$ .

Какие строки поменял местами злоумышленник?

4. Составьте для Вычислителя алгоритм проверки того, что в данной квадратной таблице равны элементы, расположенные симметрично относительно диагонали, проведенной из левого верхнего угла в правый нижний угол.

5. Пусть дана таблица наблюдений за осадками (о которой говорилось выше). Составьте алгоритм для определения:

а) самого «засушливого» месяца за один год;

б) самого «влажного» месяца за один год;

в) общего количества осадков, выпавших за один год;

г) среднего количества осадков за месяц в течение одного года;

д) числа месяцев в течение одного года, когда количество осадков превышало 40 мм.

6. Используя алгоритмы решения задачи 5 в качестве вспомогательных, составить алгоритмы определения:

а) самого «засушливого» и самого «влажного» месяца за все время наблюдений;

б) общего количества осадков, выпавших за время наблюдения;

в) наиболее «засушливого» («влажного») года.

Итог изучения алгоритмизации подводится в параграфе «Организация действий и данных при решении задач планирования». Здесь разбирается задача нахождения оптимального плана.

Учащимся предлагается представить себя в роли директора завода. На заводе организуется участок для производства товаров широкого потребления двух видов — А и Б. Для этих целей министерство выделило ресурсы (металл, электроэнергию) и определило численность работающих на участке. Кроме того, участку установлен план по реализации, указывающий минимальные объемы

производства для каждого изделия. Цель — добиться максимальной прибыли.

Сформулируем задачу конкретно.

**Задача.** На участке работают 20 человек; каждый из них в среднем за год работает 1800 ч. Выделенные ресурсы: 32 т металла, 54 тыс. кВт·ч электроэнергии. План по реализации: не менее 2 тыс. изделий А и не менее 3 тыс. изделий Б. На выпуск 1 тыс. изделий А затрачивается 3 т металла, 3 тыс. кВт·ч электроэнергии и 3 тыс. ч рабочего времени. На выпуск 1 тыс. изделий Б затрачивается 1 т металла, 6 тыс. кВт·ч электроэнергии и 3 тыс. ч рабочего времени. От реализации 1 тыс. изделий А завод получает прибыль 5 тыс. рублей, от реализации 1 тыс. изделий Б — 7 тыс. рублей. Выпуск какого количества изделий А и Б (в тыс. штук) даст наибольшую прибыль?

Учащиеся составляют математическую модель этой задачи, сводящейся к нахождению максимума линейной функции  $5x + 7y$  (прибыль) при следующих ограничениях на целые  $x$  и  $y$ :

$$x \geq 2 \quad (1)$$

$$y \geq 3 \quad (2)$$

$$3x + y \leq 32 \quad (3)$$

$$3x + 6y \leq 54 \quad (4)$$

$$3x + 3y \leq 36 \quad (5)$$

Из этих ограничений выводятся неравенства  $x \leq 9$ ,  $y \leq 8$ . При любых целых  $x$  и  $y$  ( $1 \leq x \leq 9$ ,  $1 \leq y \leq 8$ ) можно подсчитать прибыль  $V(x, y)$  по формулам:

$$V(x, y) = \begin{cases} 5x + 7y, & \text{если } x \text{ и } y \text{ удовлетворяют} \\ & \text{неравенствам (1)–(5)} \\ 0 & \text{в противном случае} \end{cases}$$

Значения  $V(x, y)$  естественно расположить в виде таблицы из 9 строк и 8 столбцов. Так возникает табличная форма организации данных. В отличие от задачи обработки метеорологических данных, в этой задаче изначально не видна необходимость в таблицах. Но в ходе решения оказывается, что за счет табличной организации удастся существенно упростить алгоритм решения задачи. По сути дела, задача сводится к нахождению максимального элемента таблицы. А этот алгоритм уже рассматривался при решении «метеорологической» задачи.

Начиная с главы 7 учащиеся знакомятся с основными средствами решения задач на ЭВМ, в частности с наиболее простым и распространенным из современных языков программирования — Бейсиком. На наш взгляд, владение языком программирования состоит в том, чтобы знать, как в нем записываются те или иные допустимые действия и как оформляются алгоритмические конструкции (которые к этому времени должны быть прочно усвоены). В отличие от учебных ис-

полнителей, использовавшихся в предшествующих главах в основном для обучения алгоритмическому подходу к решению задач, универсальные языки программирования предназначены в первую очередь для решения на ЭВМ широкого круга задач, возникающих на производстве, в науке и технике. Поскольку Бейсик выступает лишь как новое средство для записи алгоритмов (на доступном для ЭВМ языке), главенствующую роль в изучении материала главы 7 играет решение задач и выполнение лабораторных работ. Непосредственная запись программы на Бейсике осуществляется путем формального перевода алгоритма, рассчитанного на Вычислителя.

Основным содержанием главы 7 является материал четырех лабораторных работ, на каждой из которых проводятся вычислительные эксперименты, направленные на решение «прикладных» задач. Первая из них — «артиллерийская».

**Задача.** На заданном расстоянии от пушки находится стена. Известны угол наклона ствола пушки и начальная скорость снаряда. Попадет ли снаряд в стену?

Программа решения этой задачи моделирует одиночный выстрел из пушки. Поэтому «попадание» в стену может быть достигнуто только серией последовательных запусков программы при разных значениях угла наклона ствола. Учащимся предлагается вести подбор требуемого значения угла известным артиллерийским приемом «взятия в вилку», что является подготовкой к изложению метода половинного деления.

Этот метод используется при решении следующей задачи.

**Задача.** Деревянный шар плавает в ванне с водой. Какова высота части шара, находящейся под водой?

Легко понять, что задача сводится к решению уравнения  $h^2(3R-h)/3 = P/1000g$ , где  $h$  — неизвестная высота подводной части шара,  $R$  — его радиус,  $P$  — вес шара, а  $g$  — ускорение свободного падения (слева стоит объем подводной части шара, справа — объем вытесненной им воды). Знаний учащихся недостаточно для точного нахождения  $h$ , однако, применяя метод деления пополам, можно найти  $h$  с приемлемой точностью, чему и посвящена лабораторная работа 12.

На лабораторной работе 13 учащиеся закрепляют знания о вспомогательных алгоритмах и знакомятся с понятием подпрограммы-функции.

**Задача.** Для производства вакцины на заводе планируется выращивать культуры бактерий. Известно, что если масса бактерий  $x$  г, то через день она увеличивается на  $(a-bx)x$  г, где коэффициенты  $a$  и  $b$  зависят от вида бактерий. Завод ежедневно будет

забирать для нужд производства  $m$  г бактерий. Для составления плана важно знать ответ на вопрос: как изменяется масса бактерий через 1, 2, 3, ..., 365 дней (до конца года)?

Ответом на вопрос служит последовательность значений массы бактерий через 1, 2, ... дней. Обозначим их  $x_1, x_2, \dots$ .

Указать общую формулу для определения  $x_i$  через  $t$  сложно. Учащимся рассказывают, как в этом случае можно использовать рекуррентные соотношения. Действительно, по значению  $x_i$  легко определить  $x_{i+1}$ :

$$x_{i+1} - x_i = (a - bx_i)x_i - m,$$

т. е.

$$x_{i+1} = x_i + (a - bx_i)x_i - m.$$

Эти рекуррентные соотношения определяют связь между исходными данными и результатами.

Большую часть этой лабораторной работы занимает вычислительный эксперимент.

Чтобы нагляднее представить процесс изменения массы бактерий в зависимости от времени, учащимся предлагается записать первые десять результатов и изобразить их на графике, откладывая по оси абсцисс количество дней, а по оси ординат — массу бактерий.

В ходе эксперимента учащиеся убеждаются в том, что существует такой интервал значений начальной массы, при которых к концу года масса бактерий стабилизируется на одном и том же уровне. Если же взять начальную массу за пределами этого интервала, то бактерии погибнут. (Используя аппарат высшей математики, можно показать, что и в самом деле существует предельное значение, к которому стремится последовательность значений массы бактерий. В качестве задания повышенной трудности учащимся предлагается найти точное значение предела и сравнить его с результатом, полученным в эксперименте.)

При подготовке к последней лабораторной работе главы 7 рассматривается табличный способ организации данных в Бейсике. В качестве модельной взята следующая

**Задача.** Органами милиции задержан грузовик с помидорами, похищенными на овощной базе. В городе всего четыре овощных базы, каждая из них получает помидоры из своего сельскохозяйственного района. Определить, с какой базы были вывезены помидоры. Расследование осложняется тем, что помидоры на всех базах одного сорта.

Предварительно учащиеся на простейших примерах знакомятся с понятиями математического ожидания и дисперсии случайной величины, а также с критериями Фишера и Стьюдента статистических оценок гене-

ральной совокупности. Объяснение этого материала проводится в методической разработке так.

Прежде всего учащимся демонстрируются примеры случайных величин. Один из них — число шагов от школы до дома одного и того же ученика в течение нескольких дней. Ясно, что на количество шагов влияют различные внешние факторы. Оценкой для расстояния от школы до дома может служить среднее значение этой последовательности чисел.

Далее учащимся рассказывается о дисперсии как мере точности, с которой среднее значение приближает истинное значение расстояния. Если значения случайной величины обозначить  $a_1, \dots, a_n$ , а среднее арифметическое этих значений обозначить буквой  $M$ , то дисперсия вводится как среднее арифметическое квадратов разностей  $(a_1 - M), \dots, (a_n - M)$ , т. е.

14

дисперсия  $D$  равна 
$$\frac{(a_1 - M)^2 + \dots + (a_n - M)^2}{n}$$
.

Из этой формулы учащимся видно, что чем меньше дисперсия, тем меньше отличаются результаты наблюдений от своего среднего значения и тем ближе среднее значение к истинному.

Допустим теперь, что тот же эксперимент провели несколько человек. Тогда получится несколько последовательностей результатов. Можно ли узнать, какие из результатов принадлежат одному человеку? Оказывается, можно.

Математики установили, что для этого, как правило, достаточно сравнить дисперсии и средние значения. Дисперсия и среднее значение почти так же индивидуальны, как отпечатки пальцев. Если наблюдения делал один и тот же человек, то дисперсии и средние значения будут близки, если разные люди, то далеки.

Осталось выяснить: какие значения считать близкими, а какие далекими? На этот вопрос дает ответ специальный раздел математики — статистика. Оказывается, достоверность ответа зависит от числа наблюдений. Если число наблюдений от 20 до 50, то дисперсии можно считать далекими, когда отношение большей дисперсии к меньшей больше 2. Чтобы говорить о близости средних значений двух последовательностей результатов, надо найти модуль разности средних и разделить его на квадратный корень из суммы дисперсий. Если полученное число больше 0,6, то средние значения считают далекими. В том случае, когда близки и дисперсии, и средние значения, можно сделать вывод, что наблюдения почти наверняка проводились одним и тем же человеком.

Метод сравнения средних значений и дисперсий используется в самых разных обла-

стях человеческой деятельности: в медицине — для установления диагноза, в литературоведении — для определения автора произведений, в криминалистике — для розыска преступников. Он используется и для решения задачи о хищении.

В главе 8 «Символьные переменные» учащиеся знакомятся с понятием символьной переменной и операциями над символьными переменными. Здесь учащиеся сталкиваются, по сути дела, с новым для себя абстрактным понятием слова. Поэтому, так же как и в случае таблиц, целый параграф посвящен формальному определению слова и действиям над словами (соединением и выделением части).

Приведем некоторые задачи к этому параграфу.

1. а) Придумайте два слова, результат соединения которых не зависит от их порядка.  
б) Определите, какими должны быть два слова, результат соединения которых не зависит от их порядка.

2. Даны слова *информатика*, *электрификация*. Что получится, если к ним применить операцию выделения части длины 5 начиная с третьей буквы?

3. Какой длины должно быть слово, чтобы в нем можно было выделить часть длины  $l$  начиная с  $k$ -го символа?

4. Злоумышленник стер по одному «слагаемому» в каждом из следующих равенств:

- а) ... + КОЗА = СТРЕКОЗА;
- б)  $2 + 3 + \dots = 235$ ;
- в) НЕ + ... + ХОЧУ = НЕ ХОЧУ;
- г) % §§ + ... + ( ) = % §§ < > ( ) .

Восстановить пропавшие «слагаемые».

5. Алфавит племени мумбо-юмбо состоит из трех букв — ь, ы, у, расположенных в указанном порядке. Упорядочите следующие слова по возрастанию:

- ьы (вкусно)
- ъьыыыьь (мясо)
- ъьъь (вождь)
- ыыыыы (табу)
- ыьыьь (информатика)

6. Дано пятизначное число. Используя действия выделения части и соединения, составьте число, записанное теми же цифрами, но в обратном порядке (проверьте себя: из числа 12345 должно получиться число 54321).

7. а) Найдите русское слово, которое (в алфавитном смысле) больше, чем слово *пар*, и меньше, чем слово *парус*.

б) Два слова называются соседними, если не существует слов, больших меньшего из них и одновременно меньших большего. Приведите пример двух соседних русских слов.

в) Докажите, что два различных слова од-



ной длины не могут быть соседними.

8. Если натуральные числа записывать обычным образом в алфавите, состоящем из цифр, то сумма чисел не совпадает с соединением соответствующих слов (приведите пример). Как надо записывать числа, чтобы сумма чисел была равна соединению соответствующих слов?

Следующий параграф содержит описания основных команд языка Бейсик, предназначенных для работы с символьными данными. На лабораторной работе 15 «Кодирование и декодирование сообщений» решается задача кодирования и декодирования сообщений с помощью ключевой фразы.

Глава 9 посвящена остальным «школьным» пакетам прикладных программ. В ней рассказывается о базах данных, текстовых редакторах, системах поддержки электронных таблиц и, наконец, об учебном пакете «Оптима», предназначенном для решения задач линейного программирования. На лабораторных работах учащиеся получают возможность освоить эти пакеты.

Оказалось естественным начать эту главу, знакомящую учащихся с информационной технологией, с объяснения понятий «информация» и «количество информации». Нам кажется, что вводить эти довольно сложные понятия ранее нецелесообразно, поскольку они не нужны для усвоения материала, изучаемого в первых восьми главах. Объяснение строится не вполне традиционно (фактически не используется вероятностный подход Шеннона), и потому приведем его достаточно подробно.

Вначале, показывается, что «житейское» понятие информации как сведений, кого-либо интересующих, необъективно и потому не может использоваться при решении задач, связанных с автоматической переработкой информации. Кроме того, показывается, что количество информации в «житейском» смысле слабо связано с объемом текста, содержащего эту информацию. При этом понимание полученной информации сильно зависит от предварительных знаний. Скажем, телеграмма может содержать единственное слово «еду» без уточнения — кто, куда и с какой целью едет. Человек, не располагающий всеми перечисленными сведениями, не сможет правильно понять это сообщение.

Чтобы подвести учащихся к пониманию более формализованного понятия информации и ее количества, приводится пример фантастической ситуации, когда предварительные знания отсутствуют. Приведем соответствующий текст.

«Представьте себе, что в магазине «Детский мир» вы приобрели небольшой радио-

телескоп. И теперь, направив его в нужную точку звездного неба, вы с нетерпением ждете сигналов от братьев по разуму из какой-нибудь ближайшей галактики. Ваши ожидания не напрасны: ведь хорошо известно, что всякая высокоразвитая цивилизация посылает в космос радиосигналы, стремясь установить контакт с мыслящими существами других миров. Вряд ли вы успели к началу передачи и вряд ли дождетесь ее конца. В вашем распоряжении будет только небольшая последовательность сигналов. Но и в ней содержится некая информация. Однако сообщение от братьев по разуму — это не телеграмма от любимой тети. Ведь вам, скорее всего, не известна никакая предварительная информация о внеземной цивилизации. Поэтому на вопрос о том, какую информацию содержит полученная вами последовательность сигналов, наиболее полным ответом будет предъяснение самой последовательности сигналов. Итак, в данном случае информацией выступает сама последовательность сигналов; каждый новый сигнал увеличивает количество этой информации».

Теперь уже учащимся легко уяснить, что, когда речь идет об автоматической передаче информации, ее хранении и переработке, информация — это произвольная последовательность символов; каждый новый символ увеличивает количество информации.

После этого обсуждается проблема измерения количества информации, содержащейся в данной последовательности символов. Мы проводим аналогию с измерением длины или массы. Как и при измерении этих величин, нужно выбрать эталон (эталонное слово) и научиться сравнивать с ним. Прежде чем выбрать эталонное слово, надо выбрать алфавит — «материал», из которого будет сделан эталон. Обычно такой алфавит берут двухсимвольным. Например, он может состоять из цифр 0 и 1. Эталонным считается слово, состоящее из одного символа этого алфавита. Количество информации, содержащейся в этом слове, принимают за единицу; эту единицу называют битом. Сравнить с эталоном проще всего те слова, которые записаны в том же, двухсимвольном, алфавите. Количество информации, заключенной в таком слове, полагают равным его длине — ведь, как уже говорилось, каждый новый символ добавляет информацию. Чтобы измерить количество информации в произвольном слове, его кодируют (о кодировании рассказывалось в главе 8) в двухсимвольном алфавите, а затем находят длину получившегося слова. Разумеется, при таком определении количества информации оно зависит от способа кодирования. Если способ кодирования зафиксирован (а это обычно так, ког-

да канал связи автоматический), то количество информации в сообщении определяется однозначно.

Далее говорится о том, что в современных ЭВМ каждый вводимый в машину символ (буква, цифра, знак операции и т. д.) кодируется словом длины 8 в двухсимвольном алфавите, например, буква *a* кодируется словом 01100001. Для удобства введены более крупные, чем бит, единицы количества информации: байты, килобайты и мегабайты.

Приведем некоторые задачи, закрепляющие понятия информации и ее количества.

1. Сколько символов можно закодировать словами длины 8 в алфавите, состоящем из 0 и 1?

2. Юстасу необходимо передать открытым текстом следующее сообщение: «Дорогой Алек! От всей души поздравляю Вас с Новым годом! Желаю счастья, здоровья и успехов в работе. Ваш Юстас».

16

Пеленгатор определяет место передачи, если она длится не менее 3 мин. С какой скоростью (в битах в секунду) должен передавать Юстас радиограмму? (Как обычно, одна буква — один байт информации).

3. Скорость передачи информации по каналу связи меняется в течение суток по закону:

$$V = 120 \sin(t/24 + 0,3),$$

где  $t$  — текущее время (в ч), а  $V$  — скорость передачи информации (в бит/ч).

а) Какую пропускную способность должен иметь канал связи, чтобы обеспечить бесперебойную передачу информации?

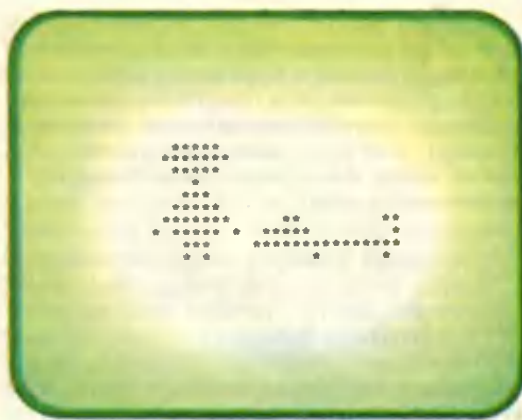
б) В какое время суток загруженность канала связи максимальна?

в) Составьте математическую модель, алгоритм и программу решения следующей задачи: определить количество информации, переданной по каналу связи в течение суток. (Здесь учащимся предлагается вспомнить задачу о равноускоренном движении тела.)

4. Какое количество информации содержится в картинке, напечатанной компьютером (рис. 11)?

В заключительной, десятой, главе рассказывается о том, как устроены компьютеры, каковы перспективы их развития и использования. В ней дается представление о системах счисления, машинных командах, памяти ЭВМ, трех основных принципах фон Неймана (программное управление, двоичное кодирование и хранение программа), о микропроцессорных средствах и системах.

В отличие от [3], мы отказались от изложения системы машинных команд реальной ЭВМ. Вместо этого, поскольку все ЭВМ устроены и работают в принципе одинаково,



для наглядности рассматривается гипотетическая «микро-микро-микро... ЭВМ» «Кроха». В ее памяти всего восемь ячеек, в каждой из которых по двенадцать разрядов. Табл. 1 — память «Крохи» в один из моментов времени. Слева от ячеек записаны их адреса (в двоичной системе счисления). Поскольку ячеек мало, то для записи адреса достаточно трех двоичных разрядов.

Таблица 1

000	0	0	0	1	0	0	0	0	0	1	1	0
001	0	0	1	1	1	0	1	1	1	0	1	1
010	1	1	1	0	0	0	0	0	0	0	0	0
011	0	0	0	0	0	0	0	0	0	0	0	0
100	0	1	1	1	0	0	1	1	0	0	0	0
101	0	0	0	0	1	0	0	0	1	1	1	1
110	0	0	1	0	1	0	1	0	1	0	1	0
111	0	1	1	0	1	1	1	0	1	1	0	1

Числа, которые можно записывать в ячейки памяти «Крохи», имеют не более 12 двоичных разрядов, следовательно, не превосходят 4095. Чтобы облегчить учащимся понимание машинных команд, мы сделали «Кроху» трех-адресной. Это позволяет не рассказывать о регистрах, о способах размещения в памяти реальной ЭВМ команд с разным количеством адресов и т. д. По нашему мнению, эти сведения неоправданно перегружают действующий курс ОИВТ.

Общий вид команды «Крохи» таков:  
В А1 А2 А3,

где В — код операции, А1, А2 — адреса операндов, А3 — адрес результата.

Вот некоторые из команд:

000 А1 А2 А3 — переписать содержимое ячейки с адресом А1 в ячейку с адресом А3 (на выполнение этой команды содержимое ячейки А2 не влияет);

001 А1 А2 А3 — сложить числа из ячеек, имеющих адреса А1 и А2, результат записать в ячейку с адресом А3;

011 A1 A2 A3 — найти модуль разности чисел из ячеек с адресами A1 и A2, результат записать в ячейку с адресом A3;

101 A1 A2 A3 — перемножить числа из ячеек с адресами A1 и A2, результат записать в ячейку с адресом A3;

111 A1 A2 A3 — остановиться и напечатать содержимое ячеек с адресами A1, A2 и A3.

«Кроха» начинает работу всегда с команды, записанной в ячейке 000. Если при выполнении какой-нибудь команды получается число, превосходящее 4095, «Кроха» отказывается продолжать работу и выдает сообщение об ошибке.

Вот примеры задач для самостоятельного решения.

1. Память ЭВМ «Кроха» заполнена следующим образом (табл. 2).

Таблица 2

```
000 0 0 0 1 0 0 0 0 0 1 1 0
001 0 0 1 1 1 0 1 1 1 0 1 1
010 1 1 1 0 0 0 0 0 0 0 0 0
100 1 1 1 1 0 0 1 1 0 0 0 0
101 0 0 0 0 0 1 0 0 0 1 1 1
110 0 0 1 0 1 0 1 0 1 0 1 0
111 0 1 1 0 1 1 1 0 1 1 0 1
```

Как будет выполняться программа?

2. Составить для ЭВМ «Кроха» программу определения:

а) периметра прямоугольника (по длинам сторон);

б) объема прямоугольного параллелепипеда (по длинам ребер);

в) половины полной поверхности прямоугольного параллелепипеда (по длинам ребер);

г) пути, пройденного телом при равноускоренном движении (исходные данные: начальная скорость, время движения и половина ускорения).

3. «Кроха» может выполнять еще одну команду: 110 A1 A2 A3 (если содержимое ячейки A1 больше содержимого ячейки A2, то перейти к выполнению команды, записанной в ячейку с адресом A3, иначе перейти к выполнению следующей команды).

Напишите для ЭВМ «Кроха» программу нахождения наибольшего из двух чисел, записанных в ячейках ее памяти.

Выбирая методические средства, мы старались сделать их оптимально соответствующими основным целям курса ОИВТ, сформулированным в начале статьи. Под оптимальностью имеется в виду разумная достаточность средств: каждый раз должны использоваться лишь такие средства, которые обеспечивают наиболее простой и короткий

путь к достижению конкретной учебной цели. Этим мы руководствовались при выборе исполнителей и средств описания алгоритмов, модельных задач и задач, иллюстрирующих применение прикладных программ, а также гипотетической ЭВМ «Кроха». Этим же объясняется порядок изложения учебного материала. Приведем поурочную разбивку курса по программе машинного варианта ОИВТ, разрешенной к использованию в 1987/88 учебном году в Свердловске и других городах области.

## Поурочная разбивка для 1-го полугодия курса ОИВТ

1—2-й уроки. Введение. Что умеет ЭВМ? Как решать задачи с помощью ЭВМ?

3—4-й уроки. Устройство ЭВМ, знакомство с клавиатурой ЭВМ. Лабораторная работа 1.

5—8-й уроки. Пример построения математической модели. Лабораторная работа 2.

9-й урок. Понятие алгоритма.

10-й урок. Исполнители алгоритмов: Чертежник и Вычислитель.

11—12-й уроки. Составление линейных алгоритмов для конкретного исполнителя.

13—16-й уроки. Ветвления. Решение задач с использованием алгоритмов с ветвлениями. Лабораторная работа 3.

17—20-й уроки. Циклы. Решение задач с использованием алгоритмов с циклами. Лабораторная работа 4.

21—24-й уроки. Решение задач на составление циклических и разветвляющихся алгоритмов. Лабораторная работа 5.

25—30-й уроки. Вспомогательные алгоритмы. Решение задач с использованием вспомогательных алгоритмов. Лабораторная работа 6.

31—32-й уроки. Решение задач с использованием ветвлений, циклов и вспомогательных алгоритмов. Лабораторная работа 7.

33—34-й уроки. Контрольная работа.

## Поурочная разбивка для 2-го полугодия курса ОИВТ

1—2-й уроки. Пошаговая детализация алгоритмов.

3—4-й уроки. Решение задач с пошаговой детализацией алгоритмов. Лабораторная работа 8.

5—6-й уроки. Организация данных.

7—8-й уроки. Табличная организация данных. Лабораторная работа 9.

9—10-й уроки. Решение задач с исполь-

зованием табличной организации данных.  
11—12-й уроки. Составление алгоритмов с использованием таблиц.

13—18-й уроки. Решение задачи планирования с использованием ЭВМ. Лабораторная работа 10.

19—20-й уроки. ЭВМ как исполнитель алгоритмов; решение задач, включающих составление линейных и разветвляющихся программ на языке программирования.

21—22-й уроки. Проведение расчетов на ЭВМ по разветвляющимся программам. Лабораторная работа 11.

23—24-й уроки. Составление циклических программ на языке программирования.

25—26-й уроки. Проведение расчетов на ЭВМ по циклическим программам. Лабораторная работа 12.

27—28-й уроки. Подпрограммы в языке программирования.

29—30-й уроки. Составление программ с использованием подпрограмм и проведение расчетов на ЭВМ. Лабораторная работа 13.

31—32-й уроки. Табличная организация данных в языке программирования.

33—34-й уроки. Контрольная работа.

18

### Поурочная разбивка для 3-го полугодия курса ОИВТ

1—2-й уроки. Составление программ с использованием табличной организации данных и проведение расчетов на ЭВМ. Лабораторная работа 14.

3—6-й уроки. Символьные переменные в языке программирования.

7—8-й уроки. Операции с символьными переменными. Лабораторная работа 15.

9—10-й уроки. Контрольная работа по составлению программ на языке программирования.

11—12-й уроки. Понятие информации и ее количества.

13—14-й уроки. Базы данных. Работа с конкретной базой данных. Лабораторная работа 16.

15—16-й уроки. Экранный редактор текстов.

17—18-й уроки. Редактирование текстов на ЭВМ. Лабораторная работа 17.

19—20-й уроки. Возможности использования электронных таблиц.

21—22-й уроки. Решение задач с помощью электронных таблиц. Лабораторная работа 18.

23—28-й уроки. Применение пакетов прикладных программ при решении задач планирования. Лабораторная работа 19.

29—32-й уроки. Функциональная организация ЭВМ. Управление памятью и внешними устройствами.

33—34-й уроки. Итоговое занятие. Роль ЭВМ в современном обществе.

Как уже говорилось, программно-методический комплекс представляет собой совокупность дидактических, методических и программных (машинно-ориентированных) компонентов. Их отработка и экспериментальная проверка в единстве и взаимодействии будет завершена в Свердловске в текущем учебном году\*. С нового учебного года он мог бы использоваться в экспериментальном порядке и в других регионах страны. Тиражирование и передачу программного обеспечения (на магнитных носителях) могут осуществлять Институт информатики АПН СССР и Свердловский пединститут. Целесообразным был бы выпуск издательством «Просвещение» подготовленного в Свердловске учебного пособия для школьников и книги для учителей тиражом, достаточным для проведения широкого эксперимента.

### Литература

1. *Ершов А. П.* Как учить программированию // Микропроцессорные средства и системы. 1986. № 1.

2. Основы информатики. Ч. 1 и 2. Методическая разработка. Свердловск, 1987. 235 с.

3. Основы информатики и вычислительной техники. Ч. 1, 2. М.: Просвещение, 1985.

\* Статья написана в октябре 1987 г.

## Проектирование программно-методического обеспечения для предметов физико-химического цикла

Наиболее серьезные психолого-педагогические исследования [1], [2], [3] и опыт использования ЭВМ в СССР и за рубежом свидетельствуют о том, что формальное внедрение компьютера в систему образования не дает ощутимых результатов. Необходимо разработать теорию и технологию проектирования программных педагогических средств (ППС), учитывающие перспективные дидактические и психологические концепции, содержание и логику предмета, методику его преподавания, конкретный тип учебного заведения.

С методических позиций предстоит разработать систему методов отображения учебной информации, а также приемов реализации выбранных методов на уровне конкретного предмета. Большую пользу в отборе и структурировании содержания может оказать технология укрупнения дидактических единиц, разработанная для математики [6].

Дидактика профессионально-технической школы на современном этапе основывается, в частности, на теории проблемного обучения [4] и концепции взаимосвязи общего и профессионального образования [5]. Это может служить методологической базой для разработки теории компьютеризации обучения. В этой связи возникает необходимость поиска критериев отбора и структурирования содержания для ППС, выделения способов управления учебным процессом, выделения методов и форм обучения, адекватных новому техническому средству.

В статье рассмотрены вопросы создания программно-методического обеспечения (ПМО) для предметов физико-химического цикла, которое состоит из двух функционально связанных подсистем:

программных педагогических средств, построенных на едином дидактическом и методическом основании и направленных на достижение определенных целей (отработка навыков алгоритмизации процессов и способов деятельности по физике, химии, формирование умений проводить эксперимент с использованием компьютера и т. п.), а также методического обеспечения, содержащего руководство по использованию этих

программных педагогических средств в учебном процессе (см. рисунок).

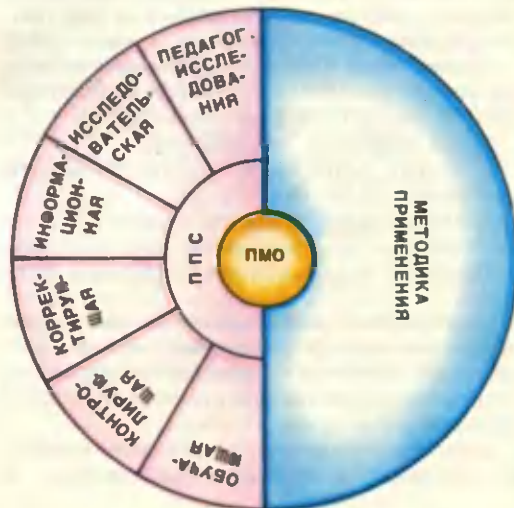
Ниже формулируются основные положения, на которых базируется разрабатываемое нами ПМО предметов физико-химического цикла для профессиональной школы.

1. Изучение возможностей и основных направлений использования вычислительной техники является необходимой частью общеобразовательной и профессиональной подготовки учащихся ПТУ.

2. Процесс компьютеризации обучения не является самоцелью. Он призван совершенствовать учебный процесс, оказывая влияние на все компоненты дидактической системы обучения, включая цели, принципы, содержание, методы, формы и характер управления процессом усвоения знаний.

3. Применение ЭВМ эффективно при условии создания целостного программно-методического обеспечения предмета с учетом особенностей, закономерностей и тенденций развития профессионально-технической школы.

4. При отборе содержания ПМО необходимо выработать систему критериев отбора содержания, подлежащего включению в ППС, в частности при представлении знаний в ППС целесообразно отбирать укрупненные понятия с целью формирования качествен-



но новых знаний, установления информационных связей между элементами знаний.

5. Систематическое использование методов научного исследования (имитационное и математическое моделирование, планирование и статистическая обработка данных эксперимента) является условием формирования современных интегративных знаний и умений.

6. Использование вычислительных и иллюстративных возможностей ЭВМ в сочетании с другими способами представления информации при создании ППС позволяет оптимально выразить сущность учебного материала, планируемого к изучению.

7. В учебном процессе ЭВМ выполняет следующие функции: собственно обучающую, контролирующую, корректирующую, информационную, исследовательскую, управляющую, а также функцию педагогического исследования. Отнесение ППС к какому-либо классу определяется его функцией на уроке.

20

8. Соответствующие типы программ являются системообразующим фактором ПМО. Технология проектирования ПМО состоит из ряда этапов.

**Первый этап.** Определение целей предмета или цикла предметов, которые необходимо учитывать при выявлении места и роли ЭВМ в процессе обучения (7), в частности:

формирование умений алгоритмизации различных процессов и способов деятельности с учетом определенных видов профессиональной деятельности;

формирование умения устанавливать взаимосвязь учебных, модельных и реальных проблем, явлений и процессов;

формирование понятий о различных способах представления информации и путях ее обработки.

**Второй этап.** Определение требований к знаниям и умениям учащихся по предметам с учетом использования в их профессиональной деятельности средств ЭВТ.

**Третий этап.** Отбор содержания учебного материала для ППС в соответствии с результатами первого и второго этапов.

Здесь возможны два подхода.

1. Отбор материала, соответствующего программе по предмету.

2. Отбор содержания, выходящего за рамки программного материала по предмету.

При этом надо учитывать следующие факторы:

установление межпредметных связей на уровне понятий, формируемых умений и методов исследования с целью создания интегративного содержания по отдельным профессионально-значимым вопросам;

активное использование методов научного исследования и приемов представления и

обработки информации в конкретной предметной области;

использование комплекса приемов укрупнения дидактических единиц [8].

**Четвертый этап.** Структурирование содержания, т. е. выделение тех дидактических единиц (законов, понятий, факторов), которые необходимы для создания ППС, и структурирование материала для разработки учебной программы конкретного типа (обучающей, контролирующей, демонстрационной и т. п.).

На этом этапе происходит установление связей между отобранным для данного ППС содержанием и методами обучения.

Выбор методов обучения, представляющих важное звено в дидактической системе, зависит от ее остальных звеньев: дидактических целей, содержания учебного материала, организационных форм, дидактических возможностей ЭВМ, а также от уровня умений учащихся работать на ЭВМ и уровня «мастерства» преподавателя в использовании ЭВМ в учебном процессе. Компьютерное обучение, несомненно, вносит значительные изменения в систему методов обучения. На данном этапе нам представляется возможным реализовать и развивать идею методов проблемного обучения [4].

Вследствие неадекватности возможностей ЭВМ современному уровню обучения, основанному на монологическом, диалогическом, эвристическом методах, реализация этих методов в компьютерном обучении приводит к примитивизации их функций. На наш взгляд, приемлемыми методами, которые разумно использовать при разработке ППС, являются методы алгоритмических предписаний, программированных заданий, исследовательский и иллюстративный. Алгоритмический метод претерпевает значительные изменения при реализации на компьютере. Он расширяет свои возможности, превращаясь из чисто репродуктивного в продуктивно-репродуктивный, при соответствующем построении учебных программ на формализуемом материале проблемного характера.

Причем его можно поднять до эвристического, если в программах предусмотрены задания по выбору оптимального алгоритма, построению алгоритма решения задачи и т. п.

**Пятый этап.** Программирование — многофункциональный процесс, объединяющий множество разнородных работ: разработку сценария учебной программы, выбор языка программирования, отладку программы, документирование и т. д.

В педагогическом отношении наиболее важной частью работ на этом этапе является разработка сценария. В сценарии учебной программы заключены содержание учебного

материала, способы представления информации на экране, характер управления процессом обучения, выбор результатов, характеризующих процесс обучения и подлежащих статистической обработке.

Сценарий должен включать: вопросы, задачи, упражнения трех уровней сложности [9];

значительный набор однотипных заданий для создания многовариантности;

несколько возможностей прохождения учебного материала для реализации адаптивности программы;

различные виды помощи, подсказок по применению законов, теорий, понятий, фактов;

различные способы представления информации для описания явлений, процессов, объектов с целью всестороннего и глубокого проникновения в их сущность и учета индивидуальных особенностей учащихся.

Шестой этап. Разработка методики применения ППС в учебном процессе, т. е. создание целостного ПМО предмета [10]. Для реализации этого этапа необходимо накопление значительного фактического материала и проведение педагогического эксперимента с целью выявления эффективности выбранного подхода к созданию ПМО для профессиональной школы.

Описанный подход к созданию ПМО реализован нами для курса химии. Для получения целостного ПМО нами проведен анализ содержания взаимосвязанных с химией предметов и выделены темы, подлежащие изучению с использованием компьютера. Причем в комплексе создаются не только готовые учебные программы, но и инструментальные средства для преподавателя.

Ниже перечислены программы\*, входящие в разрабатываемое нами ПМО для ЭВМ «Электроника ДЗ-28», КУВТ-86, ПЭВМ MSX фирмы «YAMAHA»:

комплект «Решение задач», включающий 15 типов химических задач;

программы по теме «Металлы», в состав которых входят обучающая программа «Технологические способы электролиза расплавов и растворов электролитов», исследовательская программа «Лабораторная работа по электролизу растворов и расплавов электролитов», корректирующе-контролирующая программа «Решение задач»;

полифункциональная по своему назначе-

нию программа «Окислительно-восстановительные реакции», которая по усмотрению преподавателя может быть использована для обучения, отработки навыков, контроля сформированности умений и навыков;

программа «Тепловые эффекты химических реакций», предназначенная для графического определения величин тепловых эффектов;

программы «Количественные методы химического анализа», состоящие из трех частей: самостоятельного изучения темы (в укрупненном варианте) «Титриметрия», исследовательской программы «Обработка результатов титриметрического анализа» и контролирующей программы по решению задач «Способы выражения концентрации растворов».

В настоящее время работа по созданию ПМО продолжается. Оценивается педагогическая эффективность, накапливаются статистические данные. Описанный подход используется и для создания ПМО по другим предметам.

#### Л и т е р а т у р а

1. Тихомиров О. К., Ляудис В. Ф. Психология и практика автоматизированного обучения // Вопросы психологии. 1983. № 6. С. 16—27.
2. Талызина Н. Ф. Компьютеризация обучения — научную основу // Вестник Моск. ун-та. Сер. 14, Психология. 1986. № 1. С. 3—9.
3. Машбиц Е. И. Компьютеризация обучения: Проблемы и перспективы. М.: Знание, 1986.
4. Махмутов М. И. Проблемное обучение. Основные вопросы теории. М.: Педагогика, 1975.
5. Махмутов М. И., Безрукова В. С. Принципы обучения как системообразующий фактор взаимосвязи общего и профессионального образования в среднем профтехучилище // Взаимосвязь общего и профессионального образования учащихся СПТУ. М., 1983. С. 4—21.
6. Эрднеев П. М., Эрднеев Б. П. Укрупнение дидактических единиц в обучении математике. М.: Просвещение, 1986.
7. Шакирова Д. М., Суфиярова Ф. Р. Особенности обучения химии с применением ЭВМ // Вопросы методики обучения химии и биологии в средних профтехучилищах. М., 1987. С. 54—68.
8. Шакирова Д. М. Укрупнение дидактических единиц как дидактическая основа компьютерного обучения // Укрупнение дидактических единиц. М-лы IV научно-практической конференции. 14—16 мая 1987 г. Элиста, 1987. С. 219—222.
9. Беспалько В. П. Программированное обучение. М.: Высшая школа, 1970.
10. Шакирова Д. М., Струкова Л. А. Дидактические аспекты построения программ для обучения с применением ЭВМ // Актуальные проблемы теории и практики обучения и воспитания учащихся средних профтехучилищ. М., 1986. С. 68—74.

\* Программы разрабатывались при участии сотрудников лаборатории Л. А. Струковой и Ф. Р. Суфияровой.

## Анализ содержания учебных предметов для создания педагогических программных средств

22

В настоящее время большое внимание уделяется созданию прикладного программного обеспечения для различных учебных предметов — педагогических программных средств вычислительной техники (ППС ВТ). В литературе подчеркивается, что широкое использование ППС обеспечивает повышение качества знаний учащихся, учет их индивидуальных особенностей, способствует интенсификации обучения [1, 2].

Раскроем содержание понятия *педагогические программные средства вычислительной техники*.

С одной стороны, ППС — это пакеты прикладных программ для использования в процессе обучения различным предметам. Таким образом, мы определили место ППС в составе программного обеспечения компьютера. С другой стороны, — это дидактические средства, предназначенные для различных целей обучения: формирования знаний, умений и навыков, контроля за качеством их усвоения и др., т. е. это компоненты процесса обучения. Первое принципиально важно для решения вопроса о составе и видах ППС, второе — для определения их места среди многочисленных дидактических средств обучения, выявления классов задач,

которые целесообразно решать с применением программных средств, для соотнесения традиционных и программных средств обучения.

С позиций программной реализации к числу ППС могут быть отнесены компьютерные обучающие программы (они, в свою очередь, могут быть подразделены на информационные, контролирующие, моделирующие, демонстрационные, тренажерные, комбинированные и др.), экспертные системы учебного назначения, компьютерные игры и др. Отметим, что при определенных условиях на занятиях по различным предметам могут использоваться инструментальные программные средства, такие как текстовые, графические, музыкальные редакторы, электронные таблицы и т. п.

Для каких целей в учебном процессе могут быть использованы перечисленные ППС? образное и динамическое представление учебной информации, ее систематизация, постоянное и оперативное обновление; выработка и закрепление разнообразных умений и навыков; контроль за усвоением знаний.

Этим целям можно сопоставить определенные виды программных средств (см. табл. 1).

Таблица 1

Цели применения ППС в учебном процессе	Типы ППС в порядке значимости их использования
Наглядное и образное представление, обобщение и систематизация, постоянное и оперативное обновление учебной информации	<ol style="list-style-type: none"> <li>1. Информационно-поисковые системы учебного назначения</li> <li>2. Экспертные системы учебного назначения</li> <li>3. Обучающие программы (информационные, моделирующие, демонстрационные)</li> <li>4. Компьютерные игры</li> </ol>
Выработка и закрепление разнообразных умений и навыков	<ol style="list-style-type: none"> <li>1. Обучающие программы (тренажерные)</li> <li>2. Информационно-поисковые системы учебного назначения</li> <li>3. Экспертные системы учебного назначения</li> <li>4. Компьютерные игры</li> </ol>
Контроль за усвоением знаний	<ol style="list-style-type: none"> <li>1. Обучающие программы (контролирующие)</li> <li>2. Компьютерные игры</li> <li>3. Информационно-поисковые системы учебного назначения</li> <li>4. Экспертные системы учебного назначения</li> </ol>



Широкое внедрение разнообразных ППС в учебный процесс приводит к изменению всей методической системы обучения, свидетельствует о появлении новой информационной технологии обучения [2].

Анализ практики создания и использования ППС показывает, что существуют определенные трудности и недостатки.

Во-первых, при создании педагогических программных средств не всегда уделяется должное внимание анализу и отбору учебного материала, зачастую не обосновывается необходимость и целесообразность их разработки. Разработчики часто ориентируются на создание отдельных ППС по разделу, теме, уроку без обоснования того, почему создаются именно такие ППС. В результате — фрагментарность в создании программных средств, случайный характер их использования в учебном процессе.

Во-вторых, в большинстве случаев создание программного обеспечения для учебного процесса происходит в отрыве от разработки методики его последующего применения. Анализ литературы показал, что вопросы методики применения ППС практически не рассматриваются. Поэтому преподаватель, которому предстоит использовать программное обеспечение в учебном процессе, оказывается в сложном положении: он либо вообще не знает, как рациональнее использовать ППС, либо оказывается в условиях жесткой регламентации организации учебного процесса с применением программных средств.

В-третьих, чаще всего программное обеспечение для учебного процесса разрабатывается вне связи с созданием комплексного методического обеспечения по предмету. В результате ППС либо дублируют имеющиеся дидактические средства обучения, например диапозитивы, кинофильмы, транспаранты, либо разрабатываются в отрыве от создания необходимых дополнительных средств обучения (тетрадей на печатной основе, карточек-заданий, рабочих листов и др.). И в том и в другом случае снижается эффективность использования ППС.

В-четвертых, качество создаваемых программных средств часто невысокое, по большинству учебных предметов разрабатываются однотипные по своей сущности ППС.

Для того чтобы повысить качество разрабатываемых программных средств и эффективность их использования в учебном процессе, необходимо особенно тщательно подходить к анализу и отбору учебного материала.

Рассмотрим эти вопросы более подробно. Деятельность по анализу и отбору должна проводиться на нескольких уровнях:

1. Уровень процесса обучения в целом.

2. Уровень цикла учебных предметов.

3. Уровень учебного предмета.

4. Уровень раздела (темы) учебного предмета.

Анализ на уровне процесса обучения означает формирование перечня целей для использования ППС. Причем могут быть выделены общие цели (см. табл. 1). В дальнейшем эти цели могут конкретизироваться.

На уровне цикла учебных предметов осуществляется анализ специфики содержания обучения, характерного для данного цикла. На этом этапе немаловажным представляется анализ содержания соответствующей группы наук.

Рассмотрим пример.

В профессионально-технических училищах изучаются два основных цикла учебных предметов: общеобразовательный и профессионально-технический. Такое деление не является конечным; так, в общеобразовательном цикле можно выделить гуманитарный и естественнонаучный циклы, в профессионально-техническом — общетехнический и специальный.

Для того чтобы определить специфику содержания предметов профессионально-технического цикла, нужно обратиться к анализу содержания технических наук.

В литературе подчеркивается, что в состав технического знания включаются понятия, описывающие работу и использование технических объектов, основы сходства объектов техники и технологии, их строение и функционирование, качественные свойства и количественные характеристики техники и технологии, физические и химические принципы действия технических средств [3].

Эти виды понятий представлены в содержании общетехнических и специальных предметов. Исходя из их анализа может быть сформулирован перечень ППС и определены цели их использования в обучении (см. табл. 2).

Отметим, что перечень будет неполным, если остановиться лишь на анализе содержания знаний и не выделить умения и навыки, которые необходимо сформировать у учащихся в процессе изучения соответствующего цикла предметов. В этом случае основой для анализа могут служить профессионально-квалификационные характеристики, в которых описаны требования к деятельности рабочего.

Таким образом, на уровне цикла учебных предметов формируется общий перечень ППС и перечень целей их использования.

На уровне учебного предмета происходит соответствующее уточнение перечня ППС и целей их использования, проектирование методики.

Типы ППС	Цели применения на уровне цикла предметов
Обучающие программы	Формирование знаний о физических и химических принципах действия технических средств; формирование знаний о работе технических объектов; контроль знаний.
Информационно-поисковые системы учебного назначения	Формирование знаний о сходстве технических объектов; формирование знаний об использовании технических объектов; формирование умения осуществлять выбор нужного технического объекта по заданным параметрам.
Экспертные системы учебного назначения	Формирование знаний о строении и функционировании технических объектов; формирование умений осуществлять диагностику и поиск неисправностей технических объектов.
Компьютерные игры	Формирование необходимых профессиональных умений и навыков.
Инструментальные средства	Формирование знаний о строении технических объектов.

24

Анализ осуществляется на основе рассмотрения всего содержания учебного предмета. Источником отбора содержания служат учебная программа, учебники и учебные пособия. Важно, что под влиянием создания ППС может существенным образом измениться структура учебной программы, потребуется разработка новых учебников и учебных пособий.

Среди критериев отбора учебного материала для создания ППС на этом этапе выделим два:

- 1) значимость данного учебного материала для усвоения содержания учебного предмета;
- 2) невозможность изучения учебного материала с использованием других средств обучения.

На уровне раздела и темы осуществляется отбор конкретного учебного содержания для его представления в ППС. При этом важно учитывать требуемый уровень усвоения. Источником отбора являются учебники и учебные пособия, методические пособия и ре-

комендации, научно-популярная литература.

Таким образом, от уровня к уровню происходит постепенная конкретизация в анализе и отборе учебного материала.

В дальнейшем предстоит более четко определить критерии, разработать методику отбора, т. е. определить совокупность методов, которые целесообразно использовать для анализа и отбора учебного материала при создании ППС.

#### Литература

1. *Машбиц Е.* Компьютеризация обучения: проблемы и перспективы // Информатика и образование. 1986, № 1. С. 110—127.
2. *Монахов В. М.* Новая информационная технология обучения — методологические и методические проблемы разработки и внедрения // Основные аспекты использования информационной технологии обучения в совершенствовании методической системы обучения. М., 1987. С. 3—17.
3. *Стуль Я. Е., Суханов К. Н.* Понятия технического знания и их развитие // Философские вопросы технического знания. М., 1984. С. 5—21.

## О содержании учебных компьютерных программ

С появлением ЭВМ возникла новая информационная технология, т. е. новые способы передачи и обработки информации. Любой учебный материал является в том или ином виде организованной информацией. Следовательно, принципиальных технических барьеров на пути переложения школьных курсов на ЭВМ нет. С таких позиций закладываются в компьютер взятые из учебников параграфы и примеры. Это нецелесообразное занятие прямо обусловлено неразработанностью общей концепции компьютерной технологии обучения с ее сложными психолого-педагогическими проблемами. Существует ряд общих методических вопросов, разрешение которых определяет содержание и объем учебного материала, взятого за основу при составлении сценария компьютерной программы. Данная работа ограничивается вопросами, возникающими на предсценарном этапе создания ППС — этапе формирования темы и ее содержания.

Существенная проблема на пути внедрения ППС в школу — оценка «рентабельности» программ, или, другими словами, решение вопроса, можно ли данный учебный материал при имеющихся технических возможностях раскрыть так, чтобы педагогическая эффективность компьютерного обучения была выше, чем на обычном уроке. Возникает и другой вопрос: на каком этапе обучения и каким образом компьютер становится предпочтительнее слова и действия учителя? Можно выделить четыре преимущества ПЭВМ: во-первых, возможность учебного моделирования — представления изучаемых объектов и явлений в их внутреннем развитии с созданием обучаемому условий для самостоятельного конструирования объектов и управления их динамикой; во-вторых, возможность реализации методик суггестопедии, например использование эффекта гипермнезии (сверхзапоминания) при обучении иностранным языкам; в-третьих, ЭВМ — лучшее средство для сюжетно-дидактических игр; в-четвертых, возможность индивидуализировать процесс обучения путем создания адаптивных обучающих программ.

Перечисленные потенциальные возможности компьютера должны рассматриваться

разработчиками как целесообразные пути интенсификации учебного процесса.

Актуальна и такая психолого-педагогическая проблема, как отторжение от учащегося при современных формах обучения возможности самостоятельно принимать решения. В повседневной практике школы индивидуальность подавляется слишком часто, начиная с того, что «правая рука должна лежать на левой», и кончая массой заданий типа «реши по образцу». Поэтому не случайно молодежь упрекают в безответственности и безынициативности. Изучение создаваемых ППС показывает, что та же тенденция прослеживается в компьютеризации, но в новом качестве. Жесткий алгоритм решения, слабая обратная связь и заранее предлагаемые варианты ответа — все это заставляет учащегося подстраиваться под компьютер. В идеале видится диаметрально противоположное положение, при котором человек находится не при деятельности компьютера, а внутри нее, активно вмешиваясь в эту деятельность.

Таким образом, мы приходим к необходимости формирования операционально-дидактической среды — создаваемой с помощью компьютера области, оперирующей объектами которой учащийся имитирует реальную деятельность человека, явления действительности. В операционально-дидактической среде (ОДС), оснащенной справочно-информационной подсистемой, учащийся мог бы получать необходимые для заданной предметной области умения и навыки, через практику усваивать знания и самостоятельно выбирать пути решения поставленных перед ним задач, являясь главным действующим лицом.

Разработка таких ОДС требует уже не простого копирования учебных пособий, а тщательной их переработки для того, чтобы, во-первых, создавать на основе знаний (фактов, сведений и т. п.) из учебной темы (курса) объекты ОДС; во-вторых, в соответствии с умениями (навыками), которые необходимо сформировать (развивать), построить интерфейс и процессы оперирования объектами ОДС; в-третьих, ввести в ОДС ограничения, адекват-

ные существующим закономерностям (аксиомам, физическим законам и т. д.); в четвертых, получить такой конечный результат деятельности ученика, который бы достаточно верно отражал цели обучения и реальную действительность.

26 Таким образом, желательно аккумулировать все предметное содержание учебной темы (курса) в одной программе, выполнение которой потребует от ученика знания и понимания предшествующего материала. Например, в курсе школьной биологии учебник дает декларативно связанные сведения из области биохимии, молекулярной биологии, генной инженерии и т. д. В свою очередь, компьютер может дать возможность ученику самому смоделировать процесс деления клетки. При этом возникает как минимум три положительных момента: знания, накопленные на практике, получают хорошее закрепление и обретут крепкие смысловые связи; не навязывая ученику алгоритмов мышления, мы даем ему возможность работать сообразно его индивидуальности, при этом на модель и ее динамику накладываются ограничения, определяемые только законами природы; для учителя это своего рода контрольная работа, в ходе которой проверяется не знание сведений, а понимание предмета.

Следующий вопрос, на котором необходимо заострить внимание,— это вопрос об информативности учебного материала.

Создатели компьютерных учебных программ обычно ограничены нехваткой оперативной памяти ПЭВМ, лимитами времени, установленными рамками урока и гигиеническими нормами работы с компьютером. В этих условиях при определении содержания материала важно изначально установить объем учебной информации. Действительно, большой объем обычно ведет к увеличению числа готовых формулировок и выводов, что никоим образом не стимулирует развитие творческого мышления; кроме того, известно, что даже хорошее изложение главного может утонуть в обилии частных. Поэтому в поиске золотой середины следует стремиться к тщательному отбору и объяснению узловых моментов учебной темы, быть уверенным: основное будет понято.

В заключение хотелось бы добавить, что, формируя содержание учебной компьютерной программы, следует изначально разрабатывать и методическую основу дидактического комплекса, в который должны входить как минимум сама учебная компьютерная программа, методическое пособие для учителя по ее использованию и инструкция — пособие для ученика. Такая предварительная подготовка поможет точнее определить место компьютерной программы в учебном процессе, лучше распределить учебную нагрузку в системе «учитель — компьютер — ученик» и, следовательно, повысить педагогическую эффективность разработанных программ.

О. АНШАКОВ, Н. ПИСАРЕНКО

## Методические рекомендации по использованию машинной графики БК-0010

ЭВМ БК-0010 обладает развитой системой машинной графики, которая может с успехом применяться в дидактических и игровых программах.

Написание графических программ с последующим исполнением является весьма увлекательным занятием, способным повысить интерес к работе с компьютером у

обучаемых. Нам приходилось проводить занятия по компьютерной графике со студентами Орехово-Зуевского пединститута, школьниками X классов, слушателями летних курсов переподготовки учителей информатики, преподавателями педучилищ. Их всегда интересовали подобные занятия. Дело в том, что даже при построении на экране

дисплея всем известных графиков элементарных функций (параболы, синусоиды) присутствуют элементы динамической творческой игры. Происходит конструирование графика по желанию пользователя. Применение компьютерной графики будет полезным и при изучении в школе факультативных курсов по математике, предполагающих знакомство с параметрическим заданием кривых, полярной системы координат и некоторыми замечательными кривыми.

Разумеется, при обучении машинной графике встречаются трудности. Одна из них состоит в том, что у учащихся сложился определенный стереотип в восприятии системы координат. Она предполагается прямоугольной декартовой; ось абсцисс направлена вправо, ось ординат — вверх, начало координат находится в середине листа принтера, единица измерения имеет физические размеры около 5—10 мм (1—2 клетки тетрадного листа). Система же координат экрана дисплея устроена совсем иначе: начало координат расположено в левом верхнем углу, ось ординат направлена вниз, единица измерения очень мала (около 1 мм) и имеет длину по горизонтали и вертикали.

Но, по нашему мнению, нет никакой необходимости ломать сложившееся представление системы координат при обучении машинной графике. Тем более, что во многих случаях именно привычная система координат обеспечивает наибольшую наглядность графиков.

Неудобство системы координат экрана преодолевается за счет введения так называемой рабочей системы координат. Ее выбор полностью определяется желанием пользователя БК-0010. Он задает построение в рабочей системе координат. Далее, по формулам рабочие координаты преобразуются в экранные и изображается точка с заданными рабочими и вычислительными экранными координатами.

Еще одна трудность возникает при попытке преподнести графические программы в свернутом виде. Сжатые графические программы плохо воспринимаются неподготовленным пользователем.

Для преодоления этой трудности мы предлагаем сообщать четкую формулировку графического алгоритма в виде блок-схемы, а графическую программу записывать, вычлняя отдельные блоки, даже в том случае, когда они могут быть свернуты в один оператор. Например, предлагаем отделять вычисление значений функции  $F(x)$  от изображения точки с координатами  $(x, F(x))$ . Полученные графические программы будут хоть и более длинными, но существенно более наглядными по сравнению со сжатыми программами.

Модифицируя отдельные операторы таких программ, можно легко вносить желаемые изменения в график.

Принятый в статье уровень подробности в написании программ не предел. Например, подпрограмма «построение точки с рабочими координатами  $(X, Y)$ » рассматривается нами как один оператор. Уровень подробности можно увеличить, выделив в этой подпрограмме два блока:

1) нахождение экранных координат  $(XE, YE)$ , соответствующих рабочим  $(X, Y)$ ;

2) построение точки с экранными координатами  $(XE, YE)$ .

Предлагаемые методические приемы используются ниже при изложении материала. В статье рассматривается система машинной графики БК-0010 в целом, хотя акцент сделан на построение графиков функций и параметрически заданных кривых. Излагается также построение ломаных в системе координат экрана, а также способ построения изображений с помощью оператора TYPE.

Материал излагается в логической последовательности, примеры даны в порядке возрастания трудности. На каждый рассматриваемый тип построений приводится минимум примеров (образцов программ), необходимых для понимания статьи. Учителю на уроке при работе с учащимися для закрепления материала нужно кроме приведенных в статье предлагать аналогичные примеры в качестве заданий для самостоятельного выполнения.

Авторы статьи не ставили себе задачу составить типовую программу по обучению машинной графике на БК-0010 и, тем более, дать поурочную разработку. Мы надеемся, что квалифицированный учитель информатики после ознакомления с этой статьей сможет составить самостоятельно программу и разработку.

Для понимания текста статьи необходимо знание основных операторов языка Фокал в объеме «Руководства пользователя микро-ЭВМ «Электроника БК-0010». Знакомство с машинной графикой БК-0010 не обязательно.

## Введение в машинную графику БК-0010

Для получения графического изображения на экране дисплея используются две функции: FT — формирование точки на экране и FV — формирование вектора на экране. Формат записи функций следующий:

а) FT(КОП, X, Y), где

КОП =  $\begin{cases} 0 & \text{— стирание,} \\ 1 & \text{— запись,} \end{cases}$

X, Y — координаты точки на экране;

б) FV(КОП, X, Y), где

КОП =  $\begin{cases} 0 & \text{— стирание,} \\ 1 & \text{— запись,} \end{cases}$

X, Y — координаты конца вектора.

Если начинать работу с изображения вектора, то при выполнении команды  $X \leftarrow FV(1, X, Y)$ , будет построен вектор, исходящий из верхнего левого угла экрана и заканчивающийся в точке с координатами (X, Y). Чтобы построить вектор, расположенный в любом месте экрана, необходимо задать координаты начальной точки. Это можно сделать с помощью функции FT(1, X0, Y0). Тогда последовательность команд

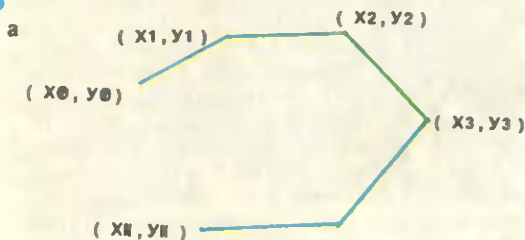
$X \leftarrow FT(1, X0, Y0);$

$X \leftarrow FV(1, X1, Y1)$

позволит изобразить вектор, имеющий начало в точке (X0, Y0) и конец в точке (X1, Y1).

Для изображения ломаной линии можно воспользоваться серией команд (рис. а)

28



$X \leftarrow FT(1, X0, Y0)$

$X \leftarrow FV(1, X1, Y1)$

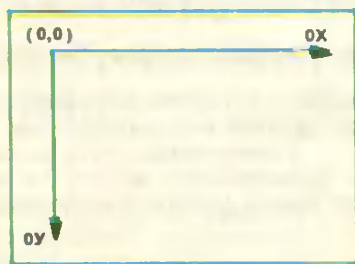
$X \leftarrow FV(1, X2, Y2)$

$X \leftarrow FV(1, XN, YN)$

В этом случае началом следующего вектора будет являться конец предыдущего.

Начало координат на плоскости экрана помещено в левый верхний угол. Положительное направление оси абсцисс — слева направо; оси ординат — сверху вниз (рис. б).

б



Работать можно с «маленьким» курсором (64 символа в экранной строке) или с «большим» курсором (32 символа в экранной строке).

Пределы изменения координат X и Y следующие:

при работе с «маленьким» курсором

$0 \leq X \leq 511,$

$0 \leq Y \leq 239;$

при работе с «большим» курсором

$0 \leq X \leq 255,$

$0 \leq Y \leq 239.$

Если значения координат X, Y выходят за границы допустимых, то машина «рисует» точку или вектор как бы за пределами экрана. Естественно, что этого изображения не видно, но функции FT и FV выполняются.

Из-за особенности экрана дисплея расстояние между двумя соседними светящимися «точками» по горизонтали и вертикали неодинаково. Поэтому длины горизонтально и вертикально расположенных отрезков, содержащих одинаковое число светящихся точек, не равны и находятся в отношении 5/4.

Во многих случаях, однако, необходимо иметь одинаковые масштабные единицы по горизонтали и вертикали. Нетрудно уравнивать масштабные единицы, умножая координату X или координату Y (или X и Y одновременно) на некоторый коэффициент (коэффициенты). Например, в случае большого курсора можно умножить координату X на 4/5 или координату Y на 5/4.

Нередко требуется сопровождать графическое изображение буквенными (символьными) обозначениями и некоторым пояснительным текстом. Чтобы правильно расположить символы по отношению к чертежу, нужно использовать тот факт, что курсор представляет собой прямоугольник, содержащий 8 светящихся «точек» по горизонтали и 10 — по вертикали. Тогда координатам точки графика X, Y будут соответствовать координаты курсора XK, YK (XK — номер позиции курсора в строке, YK — номер строки), где

$XK = [X/8],$

$YK = [Y/10].$

Перемещение курсора по плоскости экрана осуществляется с помощью функции FK(XK, YK), где в случае большого курсора

$0 \leq YK \leq 23$

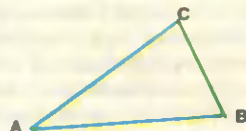
$0 \leq XK \leq 31,$

в случае малого курсора

$0 \leq YK \leq 23$

$0 \leq XK \leq 63.$

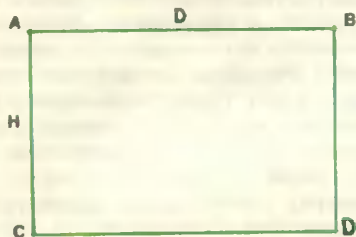
Пример 1.1. Нарисовать треугольник ABC, вершины его обозначить соответствующими буквами (рис. 1).



- 1.1 X FT(0,91,135);C точка A
- 1.2 X FV(1,133,99);C вектор AC
- 1.3 X FV(1,165,135);C вектор CB
- 1.4 X FV(1,91,135);C вектор BA
- 1.5 X FK(91/8-1,135/10);T "A"
- 1.6 X FK(133/8,99/10);T "C"
- 1.7 X FK(133/8+1,135/10);T "B"

Пример 1.2. Подпрограмма построения прямоугольника (рис. 2). Задать координаты точки A(X, Y), длину D, высоту H.

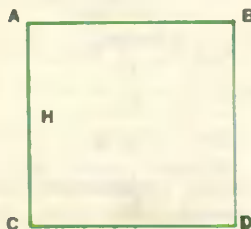
2



- 10.1 A X,Y,"D",D,"H",H
- 10.2 X FCHR(12)
- 10.3 X FT(1,X,Y)
- 10.4 X FV(1,X,Y+H)
- 10.5 X FV(1,X+D,Y+H)
- 10.6 X FV(1,X+D,Y)
- 10.7 X FV(1,X,Y)

Пример 1.3. Подпрограмма построения квадрата (рис. 3). Задать координаты точки A(X, Y), длину стороны квадрата H.

3



- 20.1 A "X",X,"Y",Y,"H",H
- 20.15 X FCHR(12)
- 20.2 X FT(1,X,Y)
- 20.3 X FV(1,X,H+Y)
- 20.4 X FV(1,X+0.8H,H+Y)
- 20.5 X FV(1,X+0.8H,Y)
- 20.6 X FV(1,X,Y)

Пример 1.4. Подпрограмма построения треугольника. Задать координаты вершин треугольника (рис. 4) X(1), Y(1); X(2); Y(2);

4

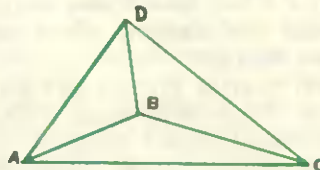


X(3); Y(3). Координаты вершин вводятся оператором A K с предварительной подсказкой на экране: X(1): Y(1):

- 30.1 F I=1,3;T !"X(",X1.01")";A X(I);
- 30.15 X FCHR(12);T "Y(",I")";A Y(I)
- 30.2 X FT(1,X(1),Y(1))
- 30.3 F I=0.2;X FV(FV(1,X(3-I),Y(3-I)))

Пример 1.5. Подпрограмма построения трехгранной призмы (рис. 5). Задать координаты вершины D (XB, YB) и координаты вершин основания X(1), Y(1); X(2), Y(2); X(3), Y(3).

5



- 40.1 A "XB",XB,"YB",YB
- 40.15 F I=1.3;A X(I),Y(I)
- 40.2 D 30
- 40.3 F I=1.3;X FT(1,XB,YB);
- X FV(1,X(I),Y(I))

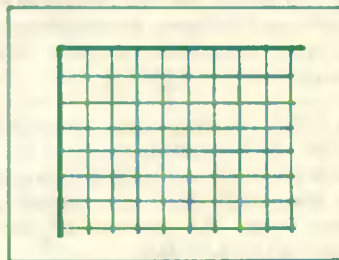
29

Пример 1.6. Подпрограмма построения произвольного N-угольника. Задать N, координаты вершин X(1), Y(1); ...; X(N), Y(N).

- 50.1 A "N",N
- 50.2 F I=1,N;T !"X(",X2.0,I,"")";A X(I);
- T "Y(",I,"")";A Y(I)
- 50.3 X FCHR(12)
- 50.4 X FT(1,X(1),Y(1))
- 50.5 F I=1,N-1;N-1;X FV(1,X(I+1),Y(I+1))
- 50.6 X FV(1,X(1),Y(1))

Пример 1.7. Подпрограмма построения масштабной сетки на экране дисплея (рис. 6). Линии масштабной сетки должны быть менее яркими, чем, например, оси координат или графики функций.

6



Задать единицу масштаба MT и смещение начала отсчета по оси OX: SX, по оси OY: SY.

- 60.1 A "MT",MT,"SX",SX,"SY",SY
- 60.15 X FCHR(12,146)
- 60.2 F I=8X,MT,255;X FT(1,I,SY);
- X FV(1,I,239)
- 60.3 F I=8Y,5XMT/4,239;X FT(1,SX,I);
- X FV(1,255,I)

## Построение графиков функциональных зависимостей

Система координат экрана непривычна для неподготовленного пользователя. Неудобно расположено начало координат (левый верхний угол), необычно направление оси координат (вниз), кроме того, единица измерения мала. Чтобы облегчить работу над графиками, целесообразно преобразовать систему координат к более привычному виду.

Оператор изображения точки запишем следующим образом:

$X \text{ FT}(1, X * MX + X_0, -Y * MY + Y_0).$

Здесь  $X$  и  $Y$ , соответственно, — абсцисса и ордината изображаемой точки в некоторой системе координат, более привычной для пользователя. Эту систему будем в дальнейшем называть рабочей системой координат.  $MX$  и  $MY$  — масштабные единицы оси абсцисс и оси ординат соответственно. Масштабная единица равна количеству светящихся точек, которое она содержит. Чтобы масштабные единицы выглядели на экране одинаковыми, отношение  $MX/MY$  должно быть равным  $4/5$ . Этого можно добиться, положив  $MY = 5/4MX$  или  $MX = 4/5MY$ . В этом случае можно предложить два варианта записи оператора изображения точки:

$X \text{ FT}(1, X * MX + X_0, -Y * MY * (5/4) + Y_0),$   
 $X \text{ FT}(1, X * MY * (4/5) + X_0, -Y * MY + Y_0).$

Знак «минус» перед  $Y$  поставлен для того, чтобы изменить направление оси ординат;  $X_0$  — абсцисса, а  $Y_0$  — ордината начала координат рабочей системы в системе координат экрана.

Таким образом, если некоторая точка имеет в рабочей системе координаты  $(X, Y)$ , то в системе координат экрана она будет иметь координаты  $(X * MX + X_0, -Y * MY + Y_0)$ .

**Пример 2.1.** Построение треугольника с вершинами  $A(1,6)$ ,  $B(8,10)$ ,  $C(3,1)$ .

Выберем рабочую систему координат с началом в точке  $(50, 180)$  в системе координат экрана. Пусть  $MX$  содержит 8 светящихся «точек»,  $MY = 5/4 MX$ .

Программа построения треугольника будет следующей:

```
1.1 S X=1;S Y=6
1.15 D 2
1.2 S X=8;S Y=10;D 3
1.3 S X=3;S Y=1;D 3
1.4 D 1.1;D 3
2.1 X FT(1, X*8+50, -Y*8*(5/4)+180)
3.1 X FV(1, X*8+50, -Y*8*(5/4)+180)
```

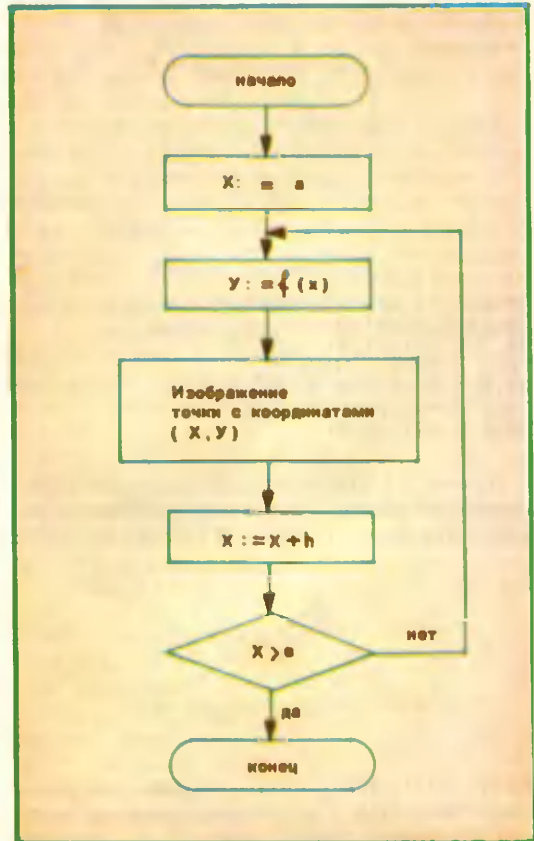
Преобразование координат удобно использовать при построении графиков функций. Графики можно строить либо с помощью

функции  $FT$ , либо с помощью функции  $FV$ . Посредством функции  $FT$  можно изобразить кривую в виде пунктира из светящихся «точек». Функция  $FV$  позволяет аппроксимировать кривую ломаной. Подобрать подходящий шаг изменения параметра, эту аппроксимацию можно сделать достаточно хорошей. Следует помнить, что в силу растровой системы машинной графики невозможно построить идеальное изображение.

Построение с помощью функции  $FV$  представляется нам предпочтительным. Изображение в этом случае может быть получено быстрее и будет более наглядным. Однако здесь мы рассмотрим построение графиков с помощью функции  $FT$ , которому соответствует более простой алгоритм.

Рассмотрим случай, когда ордината  $Y$  является функцией абсциссы  $X$ .

Для каждой конкретной задачи находим удобную рабочую систему координат, выбираем начальное и конечное значение переменной  $X$ , а также шаг изменения переменной  $X$ . Он не должен быть слишком большим. В случае большого шага мы не получим наглядного изображения кривой — на экране будут просто отдельные точки. Но шаг не





должен быть и слишком мелким. В этом случае построение будет происходить медленно. Кроме того, следует знать, что уменьшение шага не всегда приводит к улучшению качества изображения, так как вычисленные экранные координаты все равно округляются до целых. В рассматриваемом случае, т. е. когда  $Y$  есть функция от  $X$ , не имеет смысла брать шаг меньше, чем  $1/MX$ . Длина  $1/MX$  в рабочей системе координат как раз соответствует одной светящейся «точке».

Пусть зависимость между  $X$  и  $Y$  выражена формулой  $Y=f(X)$ ,  $a$  — начальное значение переменной  $X$ ,  $b$  — конечное значение переменной  $X$ ,  $h$  — шаг изменения переменной  $X$ .

Тогда алгоритм построения графика может быть описан с помощью такой блок-схемы (рис. 7).

Такой алгоритм может быть реализован на языке Фокал.

**Пример 2.2.** Построение графика функции  $y=x^2$ .

Выберем рабочую систему координат с началом в центре экрана, т. е. в точке с экранными координатами (128, 120). Пусть  $MX=10$ ,  $MY=(5/4)MX$ ,  $-3 \leq x \leq 3$ ,  $h=0,1$ .

Построение графика осуществляется по следующей программе:

```
1.1 F X=-3,0.1,3;S Y=X^2;D 2
1.9 Q
2.1 X FT(1, X*10+128, -Y*10*(5/4)+120)
```

**Пример 2.3.** Построение графика функции  $y=\sin x$ .

Пусть начало координат рабочей системы расположено в центре экрана,  $MX=10$ ,  $MY=(5/4)MX$ ,  $-2\pi \leq x \leq 2\pi$ ,  $h=0,2$  радиан.

Программа для построения графика следующая:

```
1.2 F X=-6.28,0.2,6.28;S Y=FSTN(N);D 2
1.9 Q
2.1 X FT(1, X*10+128, -Y*10*(5/4)+120)
```

**Пример 2.4.** Построение графика функции  $y=x^3$ .

Выбор рабочей системы координат, начального и конечного значения переменной  $X$ , а также шага осуществляется в диалоге с машиной.

```
1.02 A "абсцисса начала координат", XD
1.03 A "ордината начала координат", YD
1.04 A "масштаб", MX
1.05 A "начальное значение X", A
1.06 A "конечное значение X", B
1.07 A "шаг", H
1.08 X FCHR(12);C очистка экрана
1.3 F X=A,H,B;S Y=X^3;D 2
1.9 Q
2.1 X FT(1, X*MX+X0, -Y*MX*(5/4)+Y0)
```

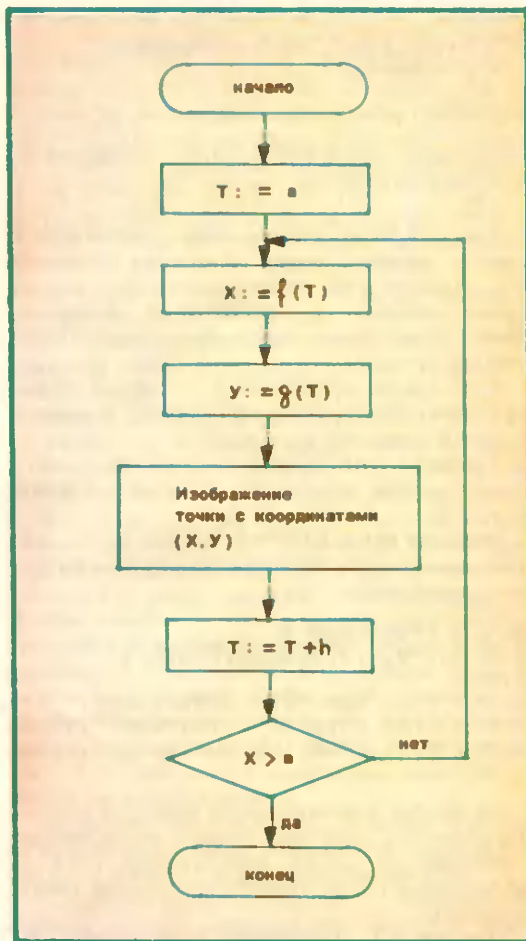
**Пример 2.5.** Построить параболу по каноническому уравнению  $y^2=2px$ .

Зададим рабочую систему координат с началом в точке (128, 120). Положим  $MX=10$ ,  $-4 \leq y \leq 4$ ,  $h=0,1$ . Значение  $p$  будем выбирать в диалоге с машиной.

```
1.05 A "параметры параболы", P
1.1 F Y=-4,0.1,4;S X=Y^2/2*P;D 2
1.9 Q
2.1 X FT(1, X*10+128, -Y*10*(5/4)+120)
```

Следует особо отметить тот факт, что в графических программах нам требуется именно функциональная зависимость. Однако канонические уравнения многих кривых не позволяют получить такую зависимость. Примерами подобных уравнений могут служить уравнения окружности ( $x^2+y^2=R^2$ ), эллипса ( $x^2/a^2+y^2/b^2=1$ ), гиперболы ( $x^2/a^2-y^2/b^2=1$ ) и многие другие. В этих случаях удобно пользоваться параметрическими уравнениями кривых, где  $X$  и  $Y$  представлены как функции некоторого параметра  $T$ .

31



Рассмотрим алгоритм построения графика с использованием параметрических уравнений (рис. 8). Пусть  $X=f(T)$ ,  $Y=q(T)$  — параметрические уравнения кривой,  $a$  — начальное значение параметра  $T$ ,  $b$  — конечное значение параметра  $T$ ,  $h$  — шаг изменения параметра.

**Пример 2.6.** Построить окружность, используя параметрические уравнения  $X = RCOST$ ,  $Y = RSINT$ .

Выберем начало координат рабочей системы в точке  $O(128, 120)$ . Положим  $MX=10$ ,  $0 \leq T \leq 2\pi$ ,  $h=0,05$ ,  $R=5$ .

1.1 F T=0,0.05,6.28;S X=5\*FCOS(T);  
S Y=5\*FSIN(T);D 2  
1.9 Q  
2.1 X FT(1,X\*10+128,-Y\*10\*(5/4)+120)

**Пример 2.7.** Построить эллипс с большой полуосью, равной 5, и малой полуосью, равной 3.

32

Начало рабочей системы координат в точке  $O(128, 120)$ ,  $MX=10$ ,  $0 \leq T \leq 2\pi$ ,  $h=0,05$ . Используются параметрические уравнения эллипса  $X=5COST$ ,  $Y=3SINT$ .

1.1 F T=0,0.05,6.28;S X=5\*FCOS(T);  
S Y=3\*FSIN(T);D 2  
1.9 Q  
2.1 X FT(1,X\*10+128,-Y\*10\*(5/4)+120)

### Некоторые дополнительные возможности машинной графики

Как уже было сказано выше, изображения кривых можно строить, используя функцию FV. Кривая при этом заменяется достаточно похожей на нее ломаной. Алгоритм такого построения описывается такой блок-схемой (рис. 9).

Блок-схема приведена для случая параметрического определения кривой. В других случаях алгоритм проще.

**Пример 3.1.** Построить окружность с центром в начале координат рабочей системы, т. е. в точке  $O(128, 120)$ .

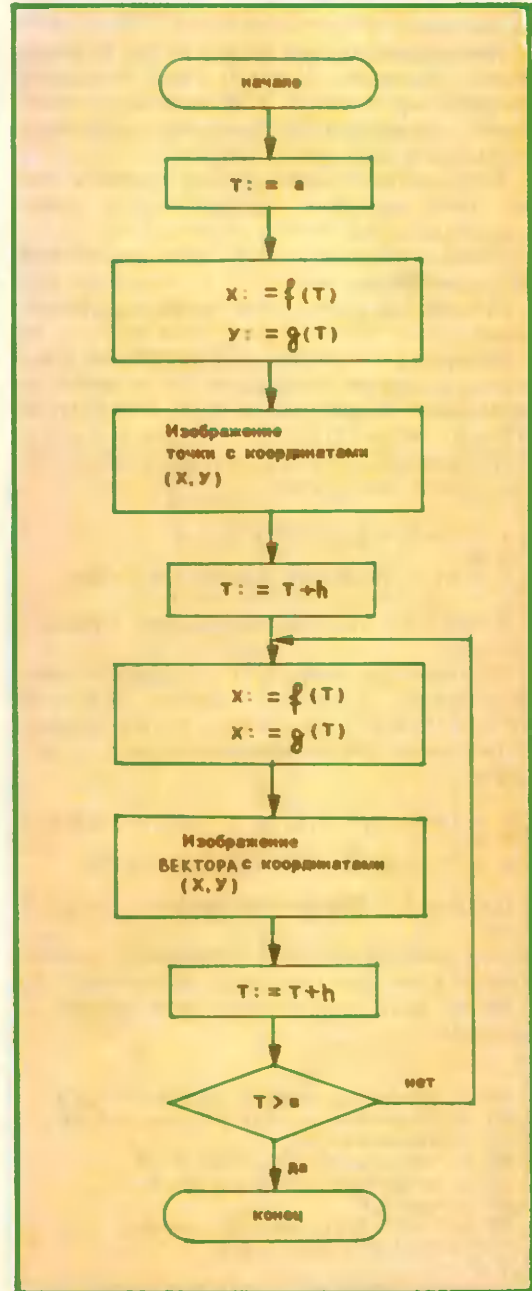
Возьмем  $MX=10$ ,  $R=5$ ,  $0 \leq T \leq 2\pi$ ,  $h=0,1$ . Тогда программа построения окружности будет следующей:

1.1 S T=0;D 2;D 3  
1.2 F T=0.1,0.1,6.28;D 2;D 4  
1.9 Q  
2.1 S X=5\*FCOS(T);S Y=5\*FSIN(T)  
3.1 X FT(1,X\*10+128,-Y\*10\*(5/4)+120)  
4.1 X FV(1,X\*10+128,-Y\*10\*(5/4)+120)

Во многих задачах удобно иметь на экране изображение осей координат с привычными стрелками, а также изображение осей и масштабной сетки. Приведем примеры таких программ.

**Пример 3.2.** Построение осей координат.

3.05 C ось абсцисс  
3.1 X FT(1,0,Y0);X FV(1,255,Y0)  
3.15 C стрелка  
3.2 X FT(1,252,Y0-3);X FV(1,255,Y0);  
X FV(1,252,Y0+3)  
3.25 C ось ординат  
3.3 X FT(1,X0,239);X FV(1,X0,0)  
3.35 C стрелка  
3.4 X FT(1,X0-3,3);X FV(1,X0,0);  
X FV(1,X0+3,3)



**Пример 3.3.** Построение осей координат и масштабной сетки.

```
1.5 B SX=XO-XOЖFTR(XO/MT)
1.55 S SY=XO-YOЖFTR(YO/MT)
1.6 X FCHR(12,146)
1.65 F I=SX,MT,255; X FT(1,I,SY);
      X FV(1,I,239)
1.7 F I=SY,MTЖ(5/4),239; X FT(1,SX,I);
      X FV(1,255,I)
1.75 X FCHR(145); D 5
```

• группа строк 5 из примера 3.2.

**Пример 3.4.** Построение графиков квадратных трехчленов. Рабочая система координат прежняя.

```
1.01 X FCHR(12) ; C очистка экрана
1.02 S XO=128;S YO=120;S MX=10;D 5
1.03 A "a",A,"b",B,"c",C
1.1 S X=-10;D 2;D 3
1.2 F X=-10+0.2,0.2,10;D 2;D 4
1.9 B 1.03
2.1 S Y=AXX^2+BXX+C
3.1 X FT(1,XXMX+XO,-YXMXЖ(5/4)+YO)
4.1 X FV(1,XXMX+XO,-YXMXЖ(5/4)+YO)
5.01 C оси координат со стрелками
5.1 X FT(1,0,YO); X FV(1,255,YO)
5.2 X FT(1,255,YO-3); X FV(1,255,YO);
      X FV(1,252,YO+3)
5.3 X FT(1,XO,255); X FV(1,XO,0)
5.4 X FT(1,XO-3,3); X FV(1,XO,0);
      X FV(1,XO,3,3)
```

Используя обычные формулы преобразования полярных координат в декартовы, нетрудно составить программы, в которых кривая задается уравнением в полярных координатах.

**Пример 3.5.** Построение спирали Архимеда ( $R=a \cdot T$ ). Рабочая система координат прежняя.  $T$  изменяется от бл,  $a=1/3$ .

```
1.1 S T=0;D 2;D 3
1.2 F T=0.1,0.1,6Ж3,14;D 2;D 4
1.9 Q
2.01 S R=(1/3)ЖT
2.1 S X=RЖFCOS(T);S Y=RЖFSIN(T)
3.1 X FT(1,XX10+128,-YX10Ж(5/4)+120)
4.1 X FV(1,XX10+128,-YX10Ж(5/4)+120)
```

### О некоторых способах иллюстрации программ графическими изображениями

Для формирования рисунков на экране дисплея кроме функций графики FV и FT можно пользоваться еще одним средством: записывать рисунок в режиме графики. Здесь предлагаются два способа записи.

**Первый способ.** Чтобы записать рисунок и в дальнейшем вызвать его на экран, нужно воспользоваться оператором TURE, а саму запись заключить в кавычки. Запись рисунка производится в режимах редактирования РЕД и графики ГРАФ. По окончании записи нужно обязательно выйти из этих режимов,

т. е. еще раз нажать клавиши ГРАФ, НР РЕД.

Остановимся подробнее на процедуре записи рисунков.

Рисунок формируется по всем правилам работы в режиме графики:

а) в режиме ГРАФ (после нажатия клавиши ГРАФ) курсор принимает вид «+» и может перемещаться по плоскости экрана;

б) в режиме ГРАФ ЗАП курсор, перемещаясь, оставляет за собой след (рисует линию). Перейти из режима ГРАФ ЗАП в режим ГРАФ можно повторным нажатием клавиши ЗАП;

в) в режиме ГРАФ СТИР курсор стирает свой след (стирает линию). Перейти из режима ГРАФ СТИР в режим ГРАФ можно повторным нажатием клавиши СТИР;

г) выход из режима ГРАФ осуществляется нажатием клавиши ГРАФ.

Перемещение курсора в выбранном направлении управляется клавишами редактирования. При нажатии одной клавиши курсор перемещается в соответствующем направлении на одну позицию. В режиме графики одна позиция передвижения курсора равна одной светящейся точке. Чтобы передвинуть курсор сразу на  $N$  позиций, совсем не обязательно  $N$  раз нажимать клавишу редактирования. Достаточно перед нажатием клавиши указать число повторений. Например, после нажатия клавиш 1, 5, 2 → курсор сразу же переместится на 152 позиции вправо.

При записи рисунка следует всегда помнить, что каждое нажатие клавиши, в том числе клавиш ГРАФ, ЗАП, СТИР, записывается во входной буфер, как передача одного кода. Поэтому от начала строки до нажатия клавиши ВВОД может быть записано не более 80 кодов.

Запись рисунков в режимах РЕД и ГРАФ наглядна, но не всегда удобна. Это связано с тем, что после выполнения оператора TURE приходится выходить из режима ГРАФ. Если требуется продолжить начатый одним оператором TURE рисунок, то, войдя заново в режим ГРАФ, мы можем не попасть именно в ту точку рисунка, где остановились. Чтобы избежать этого, можно записывать рисунки в режимах РЕД и блокировки редактирования БЛОК РЕД. Это второй способ записи графических изображений. Запись получается менее наглядной, так как передвижений курсора мы не видим. На экране вместо него возникает символическое изображение нажатой клавиши.

Например, при нажатии клавиши ГРАФ на экране появляется символ Г, при нажатии клавиши ЗАП — символ З, клавиши СТИР — С. Клавиши редактирования изображаются стрелками с соответствующим направлением.

При этом способе записи рисунка необходимо соблюдать следующую последовательность действий:

- 1) набрать T\*;
- 2) войти в режимы РЕД, БЛОК РЕД;
- 3) войти в режим ГРАФ (если это не сделано посредством предыдущего оператора ТУРЕ);
- 4) записать рисунок аналогично тому, как это делается первым способом;
- 5) если рисунок закончен, выйти из режимов ГРАФ, БЛОК РЕД, РЕД, если нет — выйти только из режимов БЛОК РЕД и РЕД;
- 6) закрыть кавычки;
- 7) нажать клавишу ВВОД.

Нетрудно заметить, что при такой последовательности действий можно завершить командную строку, не выходя из режима ГРАФ. Тогда при выполнении следующего оператора ТУРЕ автоматически устанавливается режим графики и курсор попадает точно в ту позицию, на которой прервалось исполнение предыдущего оператора ТУРЕ.

Особо рассмотрим процедуру редактирования полученных командных строк.

Если вызвать оператором MODIFY такую строку на редактирование, то на экране появится сама графическая картинка (рисунок). В таком виде строку не удастся исправить. Для того чтобы сделать исправления, нужно действовать по следующему правилу:

- 1) после приглашения «(\*)» набрать команду M(номер строки);
- 2) войти в режим БЛОК РЕД;
- 3) нажать клавишу ВВОД;
- 4) выйти из режима БЛОК РЕД;
- 5) перевести курсор в нужную позицию;
- 6) войти в режимы РЕД, БЛОК РЕД;
- 7) произвести исправления;
- 8) выйти из режимов РЕД, БЛОК РЕД;
- 9) нажать клавишу ВВОД.

Если нужно исправления сделать в нескольких местах одной строки, то каждый

раз перед очередным передвижением курсора необходимо выходить из режимов РЕД, БЛОК РЕД, т. е. повторить пункты 4—7.

Соответственно, если требуется вызвать на экран текст программы, то при включенном режиме БЛОК РЕД будут печататься символические изображения клавиш, при выключенном режиме — фрагменты рисунков.

*Пример 4.1.* Запись рисунка вторым способом (в режиме РЕД, БЛОК РЕД), используя один оператор ТУРЕ.

```
1.1. T G 70→60↓ 3 50/50\100←80↓100
→80↑ 3 70→20↓ 3 40↓40→40↑40← 3 10
→ 3 40↓ 3 20→ 3 40↑ G
```

При выполнении команды получится рис. 10

10



11

*Пример 4.2.* Запись с помощью нескольких операторов ТУРЕ.

```
2.1. T G 100↓50→ 3 30→ 32↔ 34→
↗36↔ 38→/40↔ 42→ 3 16← 3 16←↑17
→\18←↑19→/
2.2. T G 20←↑21→/22←↑23→/24←↑25→
2.3. T G 24←↑23→\22←↑21→\20←↑19
→\18←↑17→\16→↑
2.4. T G 15→\14←↑13→\12←↑11→\10
←↑9→\8←↑7→\6←↑5→\4←↑3→\2←3↑→3 ↓
3
2.5. T G 1←3↑ 3 5→\4←↑3→\2←↑2→/3
←/4→/6← G
```

После выполнения программы получится рис. 11.

34

Д. КЛИМОВ, М. ПЕТРОВ, В. УРНОВ

## «Первые шаги» и «Большие проекты»

Наверное, у каждого учителя информатики рано или поздно возникает ощущение, что приведенные в пробном учебном пособии задачи его мало удовлетворяют, причем не только в количественном плане, но и в качественном.

Во-первых, практически все они уже фор-

мализованы; во-вторых, большинство из них довольно сложны с точки зрения техники алгоритмизации и программирования. Последнее сильно затрудняет работу с учащимися, во многом способствуя возникновению барьера «человек — ЭВМ».

Мы хотим предложить несколько задач по

курсу ОИВТ, а также некоторую их классификацию, надеясь, что это не только расширит справочный материал преподавателя информатики, но и послужит полезной рекомендацией в его дальнейшей самостоятельной работе.

Вниманию читателя предлагается пять групп задач. В каждой из них есть задачи разных уровней сложности, а переход к следующей группе является, по нашему мнению, и переходом к качественно новому этапу в изучении предмета.

Первую группу составляет набор заданий, предназначенный для закрепления базовых конструкций. Назовем ее «Первые шаги». Предполагается, что эти задачи должны быть полностью формализованы, иметь очевидный (или содержащийся в формулировке) алгоритм решения и могут быть сформулированы по возможности только в терминах программирования. Подобных задач можно придумать много, и хотя в пробном учебном пособии они есть, но явно в недостаточном количестве.

Задачи второй группы представляют собой «джентльменский набор» необходимых технических приемов. Их формулировки допускают более широкий набор терминов (например, из различных школьных предметов). Однако метод и алгоритм решения по-прежнему остаются очевидными и могут потребовать от учащихся не более чем подстановки готовых формул. Такого рода упражнения можно найти во многих учебниках по языкам программирования.

Третья группа — это «Сами задачи», которые характеризуются тем, что для их решения в большинстве случаев необходимо, используя определенные технические навыки, разработать собственный алгоритм и провести частичную формализацию условия. Но здесь пока еще не требуется решать какие-либо проблемы, связанные с организацией, хранением и обработкой данных.

Четвертую группу составляют так называемые неформализованные задачи, или «Уже проблемы». Они могут быть совсем простыми и очень сложными, но главное, что для их решения необходимо найти подход, способ хранения данных, разработать метод и алгоритм. Брать такие задачи можно и нужно из повседневной жизни, из книг, сказок — откуда угодно.

Особую важность, по нашему мнению, имеют задачи пятой группы — так называемые «Большие проекты». Они подразумевают достаточно большой объем работы для группы исполнителей на продолжительное время. Деятельность такого рода позволяет научить детей многим полезным вещам: работе в коллективе, рациональному рас-

пределению обязанностей в нем, удобному и разумному планированию времени и нагрузок и т. д.

## Первые шаги

1. Заданы числа  $X$  и  $Y$ . Переменной  $M$  присвойте значение  $1$  при  $X < Y$  и  $-1$  в остальных случаях.

2. Дано число  $X$ . Переменной  $M$  присвойте значение  $1$  при  $X > 0$ ,  $-1$  при  $X < 0$  и  $0$  при  $X = 0$ .

3. Для заданного целого неотрицательного  $N$  вычислите  $N!$  ( $N! = 1 \times 2 \times 3 \times 4 \times \dots \times N$ ;  $0! = 1$ ).

4. Найдите сумму всех нечетных чисел от  $-100$  до  $100$ .

5. Найдите произведение всех четных чисел от  $-100$  до  $100$ .

6. Для исполнителя, владеющего из арифметических действий только сложением и вычитанием, составьте программу возведения целого  $A$  в целую неотрицательную степень  $N$ .

7. Измените знаки всех элементов данного массива из  $K$  чисел с четными (нечетными) номерами на противоположные.

8. В предложенном одномерном массиве замените все «нулевые» элементы единицами.

9. В массиве  $Q(X, R)$  найдите количество чисел, по модулю меньших заданного  $T$ .

10. Числовая прямая разбита на произвольные отрезки точками  $A_1, A_2, A_3, \dots, A_N$ . Выясните, какому из отрезков принадлежит данная точка  $X$ .

11. В произвольном одномерном массиве поменяйте местами элементы в парах «элемент с нечетным номером — элемент с четным номером» (дополнительные массивы не использовать).

12. Для заданного массива  $A$  из  $M$  чисел сформируйте массив  $B$  из  $N = M/2$  чисел по формуле:

$$B(I) = \frac{A(I) + A(M+1-I)}{2}.$$

Предусмотрите каким-либо образом случай, когда  $M$  нечетно.

13. Дан массив натуральных чисел  $S(N)$ . Постройте на экране дисплея  $N$  столбиков так, чтобы высота  $I$ -го столбика была равна  $S(I)$ .

14. Постройте  $N$  концентрических окружностей, радиусы которых образуют арифметическую прогрессию ( $R_1 = 1, D = \pi$ ).

15. Нарисуйте радугу, т. е. разноцветные дуги концентрических окружностей, опирающихся на нижнюю границу экрана. Толщина дуг — по вашему желанию.

16. Нарисуйте блок-схемы стандартных структур.

17. В некоторых старых ЭВМ использовалась не двоичная, а троичная система счисления. Напишите таблицу умножения для нее. Напишите также таблицу умножения для восьмеричной системы счисления.

18. Напишите программу, которая для данной функции, заданной подпрограммой, на отрезке  $(A, B)$  вычисляет сумму значений функции в  $N$  точках, делящих отрезок на  $N-1$  равных частей, умноженную на длину отрезка и деленную на  $N$  (определенный интеграл).

19. Сформируйте двумерный массив  $N \times N$  по следующему правилу: элементы главной диагонали равны 1, ниже главной диагонали — 0, а выше — сумме индексов.

20. Сформируйте массив  $C(N, N)$ , где  $C(I, J) = I! / (J! \times (I-J)!)$  при  $I \geq J$  и  $C(I, J) = 0$  при  $I < J$ .

36 21. Транспонировать произвольный двумерный массив (т. е. «поверните» его вокруг главной диагонали:  $A_{\text{новый}}(I, J) = A_{\text{старый}}(J, I)$ ). Дополнительные массивы не используйте.

22. Измените знак всех нечетных (четных) элементов массива, состоящего из  $L$  чисел (предусмотреть случай наличия нецелых элементов). Воспользуйтесь функцией INT.

23. «Сожмите» массив, «выбросив» каждый второй его элемент (дополнительные массивы использовать не разрешается).

### Джентльменский набор

24. Найдите скалярное произведение двух векторов, заданных своими координатами на плоскости.

25. Выполните задачу 24 для двух векторов в пространстве.

26. Три точки заданы своими координатами на плоскости. Найдите площадь параллелограмма, вершинами которого являются эти точки.

27. Выясните, пересекаются ли две прямые, заданные своими уравнениями на плоскости. Найдите координаты точки пересечения (если она есть).

28. Найдите площадь полной поверхности и объем правильной  $N$ -угольной пирамиды по стороне основания.

29. По заданной длине стороны правильного  $N$ -угольника найдите величину внутреннего угла, площадь, радиусы вписанной и описанной окружностей.

30. Выясните, лежат ли на одной прямой точки  $A(X_1, Y_1)$ ,  $B(X_2, Y_2)$  и  $C(X_3, Y_3)$ .

31. Решите систему двух линейных неравенств.

32. Ивана Александровича Хлестакова

пригласили управлять департаментом. В первый день ему прислали 1000 курьеров, а в каждый последующий присылали в два раза больше, чем в предыдущий. Иван Александрович согласился тогда, когда к нему прибыло сразу больше 30000 курьеров. На какой день Хлестаков согласился управлять департаментом? Учтите, что он не умеет ни делить, ни умножать.

33. Постройте на экране две касающиеся окружности, заданные своими радиусами. Рассмотрите как внешнее касание, так и внутреннее.

34.  $N$  треугольников заданы координатами своих вершин на плоскости. Найдите треугольник с наибольшей площадью.

35. Напишите подпрограмму, читающую файл страницами по  $N$  строк (в символьный массив из  $N$  элементов). Предусмотрите обработку конца файла.

36. Напишите программу, которая по требованию будет распечатывать любой из столбиков таблицы умножения (предусмотрите возможность ошибок при вводе).

37. Составьте программу, которая будет вычислять сумму нужного числа  $N$  членов данной арифметической (геометрической) прогрессии по:

а) любым двум ее членам, номера которых известны;

б) любому члену прогрессии, номер которого известен, и разности (знаменателю) прогрессии.

38.  $N$  интервалов заданы координатами своих концов на координатной прямой и произвольно пронумерованы. Задана также точка  $X$ .

А. Выясните, скольким интервалам принадлежит точка  $X$ , укажите их номера.

Б. Найдите номер и длину самого короткого и самого длинного интервалов из тех, которым принадлежит точка  $X$ .

39. Выясните, является ли предложенный массив размерности  $N \times N$  магическим квадратом (т. е. равны ли суммы его элементов по столбцам, по строкам и по обеим диагоналям).

40. Выясните, сколько раз встречается каждая буква алфавита в предложенном тексте.

41. Найдите номер максимального и минимального элементов последовательности  $A_1, A_2, A_3, \dots, A_N$ , попавших в интервал  $(A, B)$ .

42. Задан одномерный массив  $A(N)$ , состоящий только из нулей и единиц. Проверьте, строго ли они чередуются.

43. Пусть некоторая, не обязательно связанная, плоская ломаная задана координатами  $N$  своих вершин и массивом  $L(N, N)$ , где  $L(I, J) = 1$  при существовании отрезка, соеди-

няющего I-ю и J-ю вершины, и  $L(I,J)=0$  в противном случае. Постройте эту ломаную, считая, что все вершины лежат в части плоскости, ограниченной экраном;  $L(I,I)=1$ .

44. Выясните, в какой координатной четверти расположен треугольник, образованный прямой, заданной уравнением  $Y=A \times X + B$ , и осями координат.

45. Вычислите момент остановки груза,двигающегося вверх по наклонной плоскости с углом наклона  $A$ , коэффициентом трения  $K$  и начальной скоростью  $V$ .

46. Распечатайте таблицу зависимости положения тела от времени, если оно падает с высоты  $H$ ; начальная скорость  $V$ .

47. Пусть известно, что тело движется по параболическому закону:

$$S(T) = A \times T^2 + B \times T + C.$$

Напишите программу, которая по заданным положениям тела в начальный момент времени  $T_0$  и в произвольный момент  $T_1$ , а также по его скорости  $V$  в момент  $T_2$  определяла бы начальную скорость тела и ускорение.

48. Постройте прямоугольник  $A \times B$  по координатам верхнего правого угла.

49. Постройте 10 случайных прямоугольников. Воспользуйтесь функцией  $RND$ .

50. Смоделируйте движение точки (отрезка, окружности, прямоугольника).

51. Напишите программу, в процессе работы которой предварительно построенный столбик высотой в  $1/2$  экрана при нажатии на одну клавишу уменьшается на единицу длины, а при нажатии на другую — растет.

52. «Перемножьте» два двумерных массива  $A(M,M)$  и  $B(M,M)$ , т. е. сформируйте массив  $C(M,M)$ , по следующей формуле:  
 $C(1,1) = A(1,1) \times B(1,1) + A(1,2) \times B(2,1) + \dots$

$$C(1,2) = A(1,1) \times B(1,2) + A(1,2) \times B(2,2) + \dots$$

$$C(I,J) = A(I,1) \times B(1,J) + A(I,2) \times B(2,J) + \dots$$

53. Из массива  $A(N,N)$  сформируйте массив  $B(N^2)$ , «развернув» его по столбцам (по строкам).

54. Произведите операцию, обратную описанной в задаче 53.

55. Из предложенного одномерного массива длины  $N$  сформируйте двумерный массив так, чтобы первая строка нового массива содержала четные по номеру элементы исходного массива, а вторая — нечетные (предусмотрите случай нечетности  $N$ ).

56. Перепишите содержимое произвольной символьной переменной задом наперед.

57. В предложенной символьной переменной замените все кратные пробелы на одиночные.

58. Задан массив  $B(M)$ , содержащий большое количество нулевых элементов. Замените каждую группу подряд идущих нулей на один ноль.

59. В задаче 58 замените группы элементов, состоящие из нечетного количества нулей, на один нулевой элемент, а из четного — на два.

60. В задаче 58 замените каждую группу подряд встречающихся нулей на элемент, состоящий из двух чисел, где первое число — 0, а второе — количество нулей в группе.

61. В массиве  $M(N)$  осуществите сдвиг на  $K < N$  элементов вправо без использования дополнительных массивов (первые  $K$  элементов при этом остаются на месте, последние  $K$  будут потеряны).

62. В задаче 61 осуществите циклический сдвиг (т. е. последние  $K$  элементов перемещаются на первые  $K$  мест).

63. Расположите элементы произвольного одномерного массива в порядке возрастания (дополнительные массивы использовать не разрешается).

64. Заданы два одномерных массива  $A$  и  $B$ , состоящие из произвольного (не слишком большого) числа элементов. Сформируйте массив  $C$ , содержащий элементы обоих массивов, расположенные в порядке возрастания.

65. В заданном массиве  $K(N,N)$  найдите суммы элементов в каждой строке (столбце). Сформируйте одномерный массив, содержащий полученные суммы, расположенные по возрастанию (убыванию).

66. Из двух упорядоченных одномерных массивов (длины  $K$  и  $N$ ) сформируйте массив размером  $K+N$ , упорядоченный:

- а) так же, как исходные массивы;
- б) в обратную сторону.

Используйте упорядоченность исходных массивов.

67. Переформируйте двумерный массив размерностью  $N \times N$  таким образом, чтобы строки (столбцы) нового массива располагались по убыванию (возрастанию) их поэлементных сумм.

68. Имена нескольких людей (точное число неизвестно) заданы в следующем виде: кавычка, имя, по крайней мере один пробел, отчество, по крайней мере один пробел, фамилия, кавычка, по крайней мере один пробел, кавычка, имя и т. д. Напишите программу, которая считывает и печатает эти данные (одно полное имя на строке) в следующей форме: фамилия, пробел, инициал имени, точка, инициал отчества, точка.

Например, если входные данные «Модест Матвеевич Камноедов» «Федор Симеонович Киврин», то напечатано должно быть:

Камноедов М. М.

69. Напишите программу для считывания отрывка из текста, перевода его на язык Перевертыш и печати переведенного варианта. На Перевертыш слово переводится так: буквы, стоящие до первой гласной, передвигаются в конец слова и к вновь образованному слову добавляется окончание *виус*. Например:

«кот в сапогах» — «отквиус вциус апогах-свиус»

«пример перевода» — «имерпрциус ерево-дапциус»

70. Сгенерируйте текст на языке TARABARS по следующим правилам:

1) существуют специальные согласные: R, L, M, N, S, K, B, T;

2) после гласной всегда следует согласная;

3) после специальной согласной — гласная;

38 4) после простой согласной — одна из специальных согласных;

5) первая буква — любая;

6) гласные и специальные согласные встречаются в тексте в два раза чаще остальных;

7) не забудьте о знаках препинания.

71. До сих пор некоторое употребление имеет римская система счисления. Предложите эффективную процедуру перевода римских чисел в десятичные и обратно. Обозначения первых 7 римских цифр приведены в таблице.

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

72. Напишите программу, проверяющую правильность текста на языке TARABARS.

73. Предложите усовершенствования языка TARABARS, повышающие его благозвучность.

74. Выясните, сколько раз встречается каждый из символов предложенного текста.

75. Напишите программу, разбивающую произвольное предложение на слова. Распечатайте список символов, содержащихся в исходном предложении и в каждом слове в отдельности.

76. Напишите программу построения графика функции  $Y = A \sin(B \times X + C)$  в декартовых координатах с началом в центре экрана.

77. Произведите деформацию картинку, заданной двумерным массивом (1 — точка светится, 0 — нет) на экране, по закону:

$$X_{\text{нов}} = F(X_{\text{стар}}, Y_{\text{стар}}),$$

$$Y_{\text{нов}} = G(X_{\text{стар}}, Y_{\text{стар}}).$$

78. Напишите программу построения графиков функций  $Y = F(X)$ ,  $Y = G(X)$  и их суммы (разности, произведения) на одной координатной плоскости.

79. Напишите программу, которая будет определять размер файла в байтах (например, текста программы на Бейсике).

80. Найдите самое длинное звено ломаной, если заданы координаты всех ее вершин на плоскости.

81. Решите задачу 80 для ломаной в пространстве.

82. Распечатайте график функции двух переменных  $Z = \sin(X \cos Y)$  при  $0 < X < 1$ ,  $0 < Y < 1$ , представив значения по оси  $OY$  номерами строк, а по оси  $OX$  — позициями в строке; по оси  $OZ$  значение функции представлено символом, печатающимся в соответствующей позиции:

$0 < F \leq 0.1$	пробел
$0.1 < F \leq 0.2$	«:»
$0.2 < F \leq 0.3$	«.»
$0.3 < F \leq 0.4$	«=»
$0.4 < F \leq 0.5$	«+»
$0.5 < F \leq 0.6$	«  »
$0.6 < F \leq 0.7$	«X»
$0.7 < F \leq 0.8$	«C»
$0.8 < F \leq 0.9$	«Ж»
$0.9 < F \leq 1.0$	инверсный пробел

83. Аналогично задаче 82 организуйте печать графика произвольной функции двух переменных на прямоугольнике  $[X1, X2] \times [Y1, Y2]$ .

84. При покупке яблок вы заметили, что если на одну чашу весов положить  $N$  яблок, а на другую — гирю массой  $M$  кг, то яблоки окажутся тяжелее. Оцените массу одного яблока (считая их одинаковыми) в зависимости от чисел  $M$  и  $N$ .

85. С помощью подпрограммы из задачи 35 напишите программу обработки текста «Предметный указатель», которая для каждого слова из заданного списка печатает номера страниц текста, на которых слово встречается (номера не должны повторяться).

86. В заданном тексте замените все последовательности повторяющихся символов на один символ и их количество, заключенное в скобки.

87. В сказочной стране Лукоморье все тяжелые грузы перевозят на Змеях Горынычах (они имеют бортовые номера от 1 до 100). Количество голов Змея определяет его грузоподъемность.

Однажды грузовым З. Горынычем была сбита гражданка Б. Яга. Нарушитель скрылся, но свидетели показали, что бортовой номер Змея кратен его числу голов. Помогите следователю Д. Никитичу выявить потенциальных нарушителей.

88. Найдите точный минимум (или максимум) квадратичной функции на заданном интервале.



89. Выясните, пересекаются ли графики двух квадратичных функций, заданных своими уравнениями на плоскости. Найдите координаты точек пересечения, если таковые имеются.

90. Решите систему квадратных уравнений:

$$\begin{aligned} A1 \times X^2 + B1 \times X + C1 &= 0; \\ A2 \times X^2 + B2 \times X + C2 &= 0. \end{aligned}$$

91. Два отрезка заданы координатами своих концов в пространстве. Выясните, лежат ли они на одной прямой.

92. Пусть семейство из  $N$  касательных к графику некоторой функции задано для каждой касательной двумя точками — точкой касания с координатами  $X(I)$ ,  $Y(I)$  и произвольной точкой с координатами  $Q(I)$ ,  $Z(I)$ . Напечатайте таблицу значений производной функции в точках касания.

93. Напишите подпрограмму, вычисляющую значение  $I$ -го бита ( $1 \leq I \leq 8$ ) в предложенном символе (воспользуйтесь функцией ASC).

94. Произвольный выпуклый многоугольник задан координатами своих вершин на плоскости. Найдите самую длинную диагональ данного многоугольника.

95. Решите задачу 94 для многоугольника в пространстве.

96. Выпуклый многоугольник на плоскости (в пространстве) задан координатами своих вершин. Выясните, есть ли в данном многоугольнике прямые углы. Если они есть, укажите, при каких вершинах. (Предложите различные способы решения.)

97. Решите задачу 33 для трех окружностей.

98. Постройте графики многочлена  $P(X)$  степени  $N$  и касательной к нему в точке  $A(X, P(X))$ .

99. Пусть в пространстве заданы четыре произвольные точки. Определите, существует ли плоскость, их содержащая.

100. Пусть замкнутая ломаная задана координатами своих вершин на плоскости. Определите, является ли полученная фигура выпуклым многоугольником.

101. При получении отрицательного результата в задаче 100 определите, исключение каких вершин (минимального числа) позволит получить выпуклый многоугольник.

102. Вычислите площадь произвольного выпуклого многоугольника, заданного координатами своих вершин на плоскости.

103. Выясните, в одной ли полуплоскости относительно прямой, заданной уравнением  $Y = A \times X + B$ , расположены точки  $S(X1, Y1)$  и  $T(X2, Y2)$  или нет.

104. По предложенным трем числам ( $A$  — число,  $B$  — месяц,  $C$  — год) опреде-

лить день недели, соответствующий этой дате.

105. Можно ли разменять 25 рублей десятью купюрами достоинством в 1, 3 или 5 рублей?

106. Линия называется уникальной, если ее можно провести, не отрывая карандаша от бумаги. Проверьте, является ли ломаная из задачи 43 уникальной. Если да, то укажите хотя бы один путь (все пути) вычерчивания этой линии. Постройте ее.

107. Напишите диалоговую программу «Эллочка-людоедка».

108. Случайным образом перемещайте символы в строке.

109. Луч света, направленный из верхнего угла экрана по главной диагонали, падает на зеркало, установленное случайным образом, и отражается. Изобразите ход луча.

110. Постройте изображение прямого цилиндра и его образующих.

111. Постройте развертку прямоугольного параллелепипеда  $X \times Y \times Z$ , закрасив одинаковыми цветами противоположные грани.

112. Постройте развертку треугольной пирамиды, заданной длинами своих ребер.

113. Найдите все простые числа, меньшие заданного  $N$ .

114. Найдите все простые делители предложенного числа.

115. Найдите все простые числа из заданного промежутка.

116. Из одинаковых элементов двух массивов  $A(N, K)$  и  $B(S, T)$  сформируйте одномерный массив.

117. Найдите количество точек самопересечения ломаной, заданной координатами своих вершин на плоскости, укажите координаты найденных точек.

118. Решите задачу 117 для ломаной в пространстве.

119. Дан двухмерный массив  $N \times N$  ( $N$  — четное), в котором каждый элемент встречается четыре раза. «Развернув» его по строкам, удалите все повторяющиеся элементы и «сверните» обратно по столбцам в массив  $(N/2) \times (N/2)$ . Дополнительные массивы не использовать.

120. Напечатайте индексы всех совпадающих элементов произвольного одномерного (двухмерного) массива.

121. Выясните, сколько в произвольном одномерном (двухмерном) массиве содержится различных чисел.

122. Для данного одномерного массива укажите индексы тех его элементов, сумма которых равна заданному числу (если такие есть).

123. Найдите все наборы элементов, удовлетворяющих условию задачи 122.

124. Задан одномерный массив  $C(6)$ . Рас-

печатайте все различные тройки его элементов (в исходном массиве могут содержаться одинаковые элементы).

125. Решите задачу 124 для массива из  $N$  чисел.

126. Даны две строки. Выделите максимальную общую подстроку.

127. Найдите сумму длин всех общих подстрок двух данных символьных переменных.

128. Ваш Бейсик не умеет работать с трехмерными массивами. Напишите подпрограмму, имитирующую работу с такими массивами.

129. Напишите программу, имитирующую работу с  $N$ -мерным массивом.

130. Напишите программу построения графика любой функции, заданной пользователем. Предусмотрите возможность масштабирования.

131. Напишите программу построения графика функции, заданной в полярных координатах.

40 132. Постройте непрозрачное изображение прямоугольного параллелепипеда и покройте его грани различной штриховкой.

133. Постройте изображение сферы с параллелями и меридианами.

134. Постройте закрашенный треугольник по координатам его вершин и отметьте точки пересечения высот, биссектрис и медиан. Напечатайте соответствующие координаты.

135. Постройте треугольник по координатам его вершин и опишите около (впишите в) него окружность.

136. Постройте на экране окружность и отметьте на ней точку.

А. Заставьте точку двигаться по окружности.

Б. Заставьте эту окружность катиться (но не скользить!) по некоторой прямой.

137. Изобразите упругое столкновение неподвижного шара радиуса  $R_1$ , массой  $M_1$  и шара радиуса  $R_2$ , массой  $M_2$ . Последний движется со скоростью  $(VX, VY)$  из заданной точки  $(X, Y)$ .

## Уже проблемы

138. Дано неравенство  $A \times X + B > 0$ . Пользователь задает значения  $A$  и  $B$ , а также знак неравенства (подумайте, как это можно сделать). Ответ должен выдаваться в виде числового промежутка.

139. Напишите программу «Музыкальный автомат». Она должна брать строку (с клавиатуры, из файла), состоящую из текста на языке Мурзик:

Ц — целая;

П — половина;

Ч — четверть;

В — восьмая;

Ш — шестнадцатая, и «исполняет» соответствующий ритм. Не забудьте обработку ошибок!

140. Исправьте «Музыкальный автомат» так, чтобы он понимал выражения типа ПП2(Ч)Ц4(Ш)Ц, что означает ППЧЧЦШШШШЦ.

141. Исправьте «Музыкальный автомат» так, чтобы скобки с цифрой можно было вкладывать друг в друга.

142. Добавьте в Мурзик конструкцию Т (число), означающую темп (четвертей в минуту).

143. Напишите программу, генерирующую случайные тексты на Мурзике.

144. Предложите усовершенствования Мурзика. Реализуйте их.

145. Стек — это набор некоторого переменного (возможно, нулевого) числа данных, из которых доступно только последнее добавленное и не уничтоженное с тех пор, если такое существует. Это доступное данное называется верхушкой стека. Напишите программу, моделирующую работу стека. Она должна позволять:

а) поместить очередной элемент в стек;  
б) считать верхушку стека (оператор DROP);

в) обменять значения двух верхних элементов стека (оператор SWAP).

146. Напишите программу, которая получает с клавиатуры формулу одночлена (многочлена) от одной переменной (в стандартном виде) в синтаксисе Бейсика в виде символьной строки и находит его значение при заданном аргументе.

147. Перепишите программу из задачи 146 для одночлена (многочлена) от нескольких переменных.

148. Придумайте язык Курзик для управления курсором на экране.

149. Напишите интерпретатор языка Курзик.

150. Напишите подпрограмму работы с паролем, открывающим доступ к выполнению программы пользователя. Как засекретить пароль в листинге программы?

151. Напишите подпрограмму, которая получает строку символов и выводит их на экран, изображая каждую букву в матрице (например,  $4 \times 7$ ), заполняемой пробелами и символом Ш.

152. Малыш и Карлсон живут в прямоугольной комнате. Сколько им понадобится квадратных ковриков, чтобы полностью покрыть пол комнаты? (Ни Малыш, ни Карлсон не знают операций деления и умножения).

153. Напишите программу, иллюстрирующую приближенное вычисление площади криволинейной трапеции, ограниченной гра-

фиком квадратичной функции и прямыми  $X=A$ ,  $X=B$ ,  $Y=0$ . Позаботьтесь об автоматическом масштабировании.

154. Оптически прозрачная среда состоит из  $N$  анизотропных слоев со своими показателями преломления. Луч света падает на верхний слой под углом  $ALFA$ . Изобразите ход луча в среде.

155. Изобразите ход луча света в треугольной призме.

156. Утверждается, что функция  $Y=N^2 + N + 41$  является генератором простых чисел при целых  $N$ , не кратных 41. Докажите или опровергните это утверждение.

157. Постройте все несимметричные магические квадраты  $N \times N$ .

158. Вычислите число  $\pi$  с точностью 100 знаков после запятой.

159. В соревнованиях по прыжкам в длину в зачет идет лучшая их трех попыток. Для телезрителей и комментаторов постоянно необходима текущая информация о ходе борьбы спортсменов. Напишите соответствующую программу.

160. Напишите программу — переводчик с русского языка на любой другой. Переводчик разбивает текст на слова и заменяет их на соответствующие им в словаре. Дайте программе возможность обучаться, запрашивая значения незнакомых слов.

161. Миша и Вася играли в «морской бой», когда к ним подошел Дима, недавно купивший новый бытовой персональный компьютер. По окончании игры они решили написать программу, которая на одном поле строит Мишины «корабли» и симметрично отраженные относительно главной диагонали — Васины. Однако, несмотря на все усилия, у них ничего не получилось. Помогите ребятам написать такую программу.

162. В заданном тексте проверьте правильность расстановки круглых, квадратных и фигурных скобок.

163. Напишите программу, генерирующую случайный синтаксически правильный текст случайной длины на каком-либо языке программирования (например, на Бейсике).

164. Напишите программу «Дедал», генерирующую проходимые лабиринты.

165. Напишите программу «Ариадна» прохождения через лабиринты «Дедала».

166. Придумайте функцию, определяющую «похожесть» двух строк. Одна из возможных идей содержится в задаче 127.

167. В заданном тексте найдите наиболее часто встречающуюся последовательность символов заданной длины.

168. Смоделируйте работу со строками, длина которых превышает возможности вашего транслятора.

169. Даны две строки, в каждой из кото-

рых содержится очень длинное целое число. В третьей строке сформируйте их сумму (произведение).

170. Упорядочите список учеников по алфавиту. Оцените порядок количества необходимых операций.

171. Укажите подмножество реализованного на вашей ПЭВМ Бейсика, равное ему по возможностям.

172. Разработайте способ шифровки и дешифровки информации в текстовых файлах.

173. Напишите программу, которая определяет относительную частоту встречаемости операторов в заданной программе на Бейсике.

Например:  
IF — 1 %  
LET — 1 %  
GOTO — 98 %

174. Греко-латинским квадратом порядка  $N$  называется квадрат  $N \times N$ , в каждой строке, каждом столбце и обеих диагоналях которого содержатся все целые числа от 1 до  $N$ .

Пример квадрата 4-го порядка:

1	2	3	4
4	3	2	1
2	1	4	3
3	4	1	2

Напишите программу, которая:  
а) строит хотя бы один квадрат порядка  $N$ ;

б) строит все квадраты порядка  $N$ ;

в) строит все квадраты порядка  $N$  так, что нельзя получить один из другого при помощи поворотов и вращений вокруг осей симметрии.

## Большие проекты

175. Задачи 139—144 как одна задача.

176. Смоделируйте работу калькулятора МК-54 (БЗ-34):

а) в автоматическом режиме;

б) в программном режиме.

177. Напишите программу, моделирующую опыт Миллекена.

178. Напишите контролируемую программу по теме «Построения циркулем и линейкой». Программа должна сама генерировать задания.

179. Напишите обучающую программу для пятиклассников по теме «Координатная плоскость».

180. Точечный заряд  $Q$  пролетает через электрическое поле, образованное тремя произвольными зарядами  $Q_1$ ,  $Q_2$  и  $Q_3$ . Изобразите траекторию его движения.

181. Используя процедуры ДОМ, ДЕРЕВО, КУСТ, ЗАБОР, СТОЛБ и другие, нарисуйте лицо по заказу пользователя.

182. Напишите программу «Процессор»,

которая читает текст, состоящий из чисел и знаков операций (+, —, ×, /), и реализует следующий алгоритм:

1) если прочитано число, то оно помещается в стек;

2) если прочитана операция, то верхние два элемента стека заменяются результатом этой операции над ними.

Учтите некоммутативность операций вычитания и деления.

183. Нарисуйте действующую модель ча-

сов с кукушкой. Часы должны тикать, кукушка куковать.

184. Разработайте систему «БИБЛИОТЕКАРЬ».

185. Создайте для своих учителей «электронный журнал». Организуйте сбор статистики. Не забудьте про анкетные данные и учет посещаемости. Всячески облегчите поиск необходимой информации.

186. Напишите обучающую систему по теме «Сечения многогранников».

## ПОСЛЕ КОНКУРСА

Г. КОПЫЛОВ

# Из рукописи конкурсного учебника по информатике

В соответствии с решением конкурсной комиссии мы начинаем публикацию фрагментов рукописей конкурсных учебников по информатике.

## 1. Введение

### 1.1. Информация и информатика

Под информацией в широком смысле слова понимается все, что может помочь ответить на какой-либо вопрос.

Как позвонить в библиотеку? Информацию об этом (в данном случае — номер телефона библиотеки) мы можем взять из телефонного справочника. Сколько сейчас времени? Как вас зовут?

В узком смысле слова под информацией понимается содержимое книг, газет, журналов и т. д.

Человек получает информацию с помощью всех органов чувств.

Недостаточно получить информацию — необходимо суметь ее переработать. Так, надпись на японском языке ничего не скажет человеку, не знающему японского языка. Герои Конан-Дойля Шерлок Холмс и доктор Ватсон получали одну и ту же информацию, но по-разному ее обрабатывали. «Вы просто смотрите, — говорил Шерлок Холмс, — а я вижу».

Осматривая больного, выслушивая его сердце, изучая результаты анализов, врач получает информацию о больном и болезни. Анализируя ее с использованием своих зна-

ний, своего опыта, советуясь с коллегами и заглядывая в справочники, врач перерабатывает информацию и ставит диагноз, назначает лечение.

Важно уметь сохранить получаемую информацию. Человек может хранить увиденное и услышанное в своей памяти. Однако этот способ ненадежен. Часть информации может забыться, потеряться.

Информацию необходимо не только хранить и обрабатывать, но и уметь передавать. Необходимость запоминать, передавать и обрабатывать информацию привела к появлению языка, письменности, развитию книгопечатания, появлению газет и журналов, радио и телевидения. В XX в. количество накопленной человечеством информации стало столь велико, что появилась специальная наука — информатика, предмет которой — законы и методы накопления, обработки и передачи информации. В связи с появлением электронно-вычислительных машин (ЭВМ) информатика сильно развилась и изменилась. Мы будем использовать следующее более узкое определение: информатика — наука о методах переработки информации с помощью ЭВМ или, более коротко: информатика — наука об ЭВМ.

### 1.2. Исполнитель

Мы живем в эпоху научно-технической революции. Одной из характерных ее особенностей является быстрый темп изменения жизни. Можно ли представить сегодняшнюю жизнь без электричества в вашем доме,

без автомобилей и тракторов, без телевизора и магнитофона? А ведь сравнительно недавно ничего этого не было. У человека появляются все новые и новые помощники, которые могут исполнить те или иные его приказы. Таких помощников мы будем называть исполнителями.

Исполнитель — человек, организация или устройство, способное выполнить некоторые наши просьбы, приказы, команды.

Если исполнитель — человек, то просьбу или приказ можно изложить на обычном человеческом языке, понятном исполнителю, — русском, английском и т. д. Если же исполнитель — устройство, то обращаться к нему на человеческом языке бесполезно, необходимо использовать какой-то особый язык.

Приведем примеры.

Если в комнате стало темно, то можно сказать: «Лампочка, зажгись!» Однако на приказ голосом лампа не отреагирует. Чтобы лампа зажглась, необходимо ее включить. Итак, здесь команда отдается с помощью выключателя. Если человек едет в такси, то команда шоферу подается на человеческом языке. Если же человек сам ведет автомобиль, то команды он отдает с помощью руля, педалей и кнопок, имеющихся в машине. Чтобы подняться на лифте на 7-й этаж, мы даем соответствующую команду, нажимая кнопку с цифрой 7.

Подчеркнем главное.

1. Существуют автоматы-исполнители, помогающие человеку, облегчающие его жизнь, расширяющие его возможности.

2. Для общения с ними необходимо знать, какие команды они понимают и что произойдет в результате выполнения той или иной команды.

Достаточно ли иметь исполнителя и уметь им управлять, чтобы решать появляющиеся перед нами задачи? Нет. Если мы имеем автомобиль, умеем им управлять, но не знаем, куда (или зачем) ехать, то никуда и не приедем. Таким образом, очень важным является следующее условие.

3. Мы должны знать, какую задачу необходимо решить и как ее можно решить с помощью исполнителя.

Под задачей здесь понимается любая проблема, которую надо решить, любая цель, которую надо достичь: построить дом, испечь пирог, решить уравнение, посадить семена, собрать урожай и т. д.

Задание. Приведите примеры исполнителей. Какие команды понимают и могут выполнить эти исполнители? Каким образом эти команды понимает исполнитель? С какими исполнителями имеет дело ваши родители на своих рабочих местах? Кем ты хо-

чешь работать после окончания школы и какие исполнители будут помогать тебе в работе? Может ли существовать универсальный исполнитель, который выполнит любую команду?

### 1.3. Описание исполнителя Карандаш

При работе с исполнителями возможны две принципиально разные ситуации: исполнителя нет, и мы должны его придумать, т. е. сказать, какие команды он должен уметь выполнять;

исполнитель уже создан, и мы должны изучить, что он может и как его использовать для решения интересующих нас задач.

Давайте придумаем исполнителя Карандаш для построения чертежей и простых рисунков. Какие команды он должен уметь выполнять? Конечно, команду проведения отрезка. При этом надо как-то уметь задать, какой отрезок необходимо нарисовать. Как это сделать? Отрезок можно задать, указав его начало и конец. Для этого необходимо уметь задавать точки. Задать точку можно, указав ее координаты.

Итак, хорошо бы, чтобы наш исполнитель умел выполнять команду

Отрезок (А, В, С, Д)

и выполнять так: он должен соединить отрезком точку с координатами (А, В) и точку с координатами (С, Д). А, В, С, Д здесь — какие-то числа.

Давайте с помощью такого исполнителя нарисуем квадрат со стороной 3, левая нижняя вершина которого расположена в начале координат, а стороны параллельны осям координат. Это можно сделать с помощью следующей программы.

Отрезок (0, 0, 0, 3)

Отрезок (0, 3, 3, 3)

Отрезок (3, 3, 3, 0)

Отрезок (3, 0, 0, 0)

Выполнив ее, исполнитель нарисует требуемый квадрат.

Программа — совокупность команд, понятных исполнителю, выполнение которых в определенном порядке приведет к решению поставленной задачи.

Задание. Выполните следующую программу.

Отрезок (1, 2, 1, 5)

Отрезок (1, 5, 3, 5)

Отрезок (3, 5, 3, 3)

Отрезок (3, 3, 1, 3)

Что получилось при выполнении этой программы? Можно ли получить этот рисунок с помощью какой-либо другой программы?

Задание. Составьте программу написания своих инициалов.

Задание. Составьте программу написания даты своего рождения.

Задание. Нарисуйте домик с помощью исполнителя Карандаш.

Вопрос. Изменится ли результат выполнения программы, если изменить порядок команд?

Какие рисунки мы можем нарисовать с помощью исполнителя Карандаш, рассмотренного на предыдущем уроке? Любые ли?

С его помощью можно нарисовать любой рисунок, состоящий из отрезков, но нельзя нарисовать, например, окружность.

Какую команду добавить к понятным для Карандаша, чтобы с его помощью можно было рисовать окружность и дуги окружностей? Что нужно указывать в такой команде?

44 Один из возможных вариантов — научить Карандаш понимать и исполнять команду Дуга ( $A, B, R, \alpha, \beta$ )

По этой команде исполнитель должен нарисовать дугу окружности с центром в точке с координатами  $A$  и  $B$ , радиусом  $R$ ; числа  $\alpha$  и  $\beta$  задают углы, составленные с положительным направлением оси  $Ox$  радиусами, проведенными в начало и конец дуги. Величины  $\alpha$  и  $\beta$  будем указывать в градусах. При этом исполнитель будет рисовать дугу в направлении против часовой стрелки от  $\alpha$  до  $\beta$ .

При  $\alpha = \beta$  исполнитель должен нарисовать всю окружность.

Вопрос: что нарисует каждая команда?

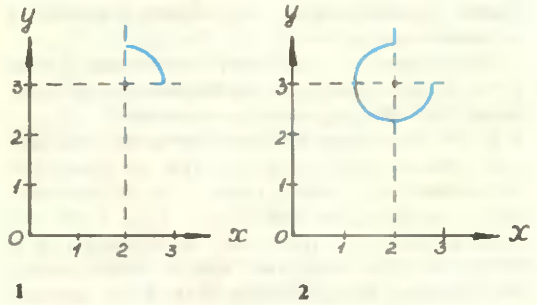
а) Дуга (2, 3, 1, 0, 0)

Ответ: окружность с центром в точке (2, 3) и радиусом 1.

б) Дуга (2, 3, 1, 0, 90)

Ответ: четверть дуги этой окружности от угла в  $0^\circ$  до угла в  $90^\circ$  (рис. 1).

в) Дуга (2, 3, 1, 90, 0)



Ответ: три четверти окружности (рис. 2).  
Вопрос: что нарисует следующая программа?

Отрезок (0, 0, 1, 1)

Отрезок (1, 0, 1, 3)

Дуга (1, 2, 1, 90, 270)

Если вы все сделали правильно, то у вас получилась буква Я.

До сих пор мы писали по одной команде в каждой строке. Переход на другую строку означал конец одной команды и начало следующей. Для экономии бумаги и большей наглядности иногда удобнее помещать в одной строке несколько команд. Но тогда нужен какой-либо знак для обозначения конца команды. Обычно для этого используется точка с запятой.

Вопрос: что нарисует следующая программа?

Отрезок (0, 1, 2, 3); Отрезок (2, 3, 2, 0);

Отрезок (2, 0, 0, 1)

Окончание следует.

### Уважаемые читатели!

Не забудьте подписаться на журнал «Информатика и образование» на 1989 г. Подписка принимается без ограничений в любом отделении связи. Индекс журнала 70423.

А. АРХАНГЕЛЬСКИЙ

## Мир ЭВМ

### 6. Файловая система

#### 6.1. Все может случиться, или Что нужно для работы

— Здравствуйте, Александр Геннадьевич!  
А что это сегодня так тихо и пусто?

— Электричество отключили, вот они и не работают.— Александр Геннадьевич взглядом показал на машины.— Так что, к сожалению, ничем помочь сегодня не могу.

Ну что ж! Работая с программами DIR и DUMP, мы получили распечатки; пойдем домой и будем в них разбираться. Ведь программисту не всегда нужна ЭВМ, иногда полезнее удобное кресло и спокойная обстановка.

Вот распечатка каталога удаленных файлов в восьмеричной форме. Ее мы получили командой

```
DIRECTORY/OCTAL/PRINT/DELETED  
DX0:
```

Она нам пригодится попозже.

А вот еще распечатка каталога, самая полная — с указанием свободных областей, начальных блоков и указанием идентификатора тома также в восьмеричной форме. Ее мы получали командой

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

Еще у нас были дампы.

Распечатку начальных блоков монитора мы получали командой\*

Продолжение. Начало см.: Информатика и образование. 1987. № 1—3; 1988. № 3.

\* Те распечатки, которые не были приведены в журнале, читателям предлагается получить самостоятельно.



```
DUMP/RAD50/END:6 DX:RTIISJ.SYS
```

Распечатку драйвера DX.SYS — командой  
DUMP/RAD50 DX:DX.SYS

Наконец, у нас есть распечатки первых восьми блоков системного тома (так громко мы называем нашу единственную дискету).

#### 6.2. Как устроен том

Мы так долго занимались «глазами» системы, что у нас зарябило в глазах. Каталоги всех видов, дампы... Попробуем применить все это на практике.

Начнем плясать от печки. Впрочем, ее нужно еще отыскать. Файловая система организована на томе, поэтому за основу возьмем распечатку дампа первых восьми

## BLOCK NUMBER 00000

```

000/ 000240 000413 000000 000000 000000 000000 000120 000340 * .....P.D.*
      D      FX      B      EX
020/ 000070 000340 000000 041420 116020 000400 116067 000056 *B.D....C....7...*
      AP      EX      J/H     X82    FP    X91    AF
040/ 000066 011706 012702 000200 005000 000446 000000 013407 *6.F.B.....&.....*
      AN      CFV     CSJ     CH     AX     GN     C#D
060/ 000000 000000 000000 000000 005714 001776 100762 000002 *.....L.4.P...*
      A5T     YV     T1Z     B
100/ 000000 000000 000000 000000 000000 000000 000000 000000 *.....*

120/ 012704 177170 010405 012725 000000 000004 010315 000004 *D.b4..U....M...*
      CSL    :X    B.7    CS/    D      B#U     D
140/ 010015 000004 012714 000003 000004 105714 100350 111522 *...L.....L.X.R.*
      BV/    D      CST     C      D      UN.     TY     W44
160/ 005301 003372 005002 000770 010601 005200 011703 000003 *A.3...b....C...*
      A.3     ABZ     AXB     LX     B2A     A#H     CFS     C
200/ 122323 000003 122323 000003 032767 000020 177712 001173 *S#...S#...B5...J...*
      ZN5     C      ZN5     C      HYD     P      :7B     05
220/ 005000 000571 012737 000167 000150 012737 000214 000152 *...W...B.X.....Я.*
      AX      10     CS9     B9     BX     CS9     CT     BZ
240/ 012737 000300 004730 012737 000416 000020 005037 000044 *...E.X.....x.*
      CS9     D2     AW     CS9     FO     F      AX1     6
260/ 005767 000346 001405 013703 004722 116367 000056 177630 *B.#...C.R.B.....*
      A6W     E0     SM     C2C     AV4     YDW     AF     :52
300/ 006300 006300 006301 010046 010003 010004 005000 000402 *E.E.A.&.....*
      BAX     BAX     BAY     BWN     BVS     BVT     AX     FR
320/ 162703 000027 005200 162704 000032 100372 022704 177762 *CE....DF...3.D/P.*
      60S     W      A#H     60T     Z      TYR     FA.     :BB
340/ 006103 162703 000032 100375 062703 000033 000003 012600 *C.CE...#.CE.....*
      ABS     60S     Z      TYU     PKK     x      C      C0X
360/ 005200 005701 003350 000207 005714 001776 100533 105714 *...A.X...L.4.L.L.*
      A#H     ASI     ADH     CD     A5T     YU     T#5     UN.
400/ 100011 111522 005301 003370 012702 000001 000765 005714 *...R.A.b.B...Y.L.*
      TS0     W44     A.3     ADX     CSJ     A      LU     A5T
420/ 100521 001775 000002 000000 000000 000000 000000 000000 *G.#.....*
      T#Y     YU     B
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 012706 010000 010046 012700 000002 *.....F...&.E...*
      CSN     BVP     BWN     CSH     B
620/ 012701 002000 012702 001000 005027 000001 004767 177364 *A...B.....B.T#*
      CSI     YX     CSJ     L2     AXW     A      AW1     :1.
640/ 012737 000300 004730 012737 016300 004716 012637 004722 *...E.X...E.N...R.*
      CS9     D2     AW     CS9     BX     AVO     CRO     AV4
660/ 000137 001000 004167 000002 000766 012700 000746 004167 *...B...B.E.#.B.*
      BD     L2     ANG     B      LV     CSH     LF     ANG
700/ 000030 012100 004167 000022 012700 000762 004167 000012 *...E.B...E.P.B...*
      X      CIX     ANG     R      CSH     LR     ANG     J
720/ 000005 000000 000776 112037 177566 105737 177564 100375 *....4...I...T.M.*
      E      LO     W#9     :4B     VGG     :46     TYU
740/ 105710 001371 000201 005015 041077 047517 026524 026525 *H.Y.....?BOOT-U-*
      UNX     SA     CI     AXM     J#9     L#W     GJD     GJE
760/ 000200 005015 000012 027511 020117 071145 067562 000162 *.....I/O ERROR.*
      CH     AXM     J      GUY     EF1     RLE     Q3J     B4

```

блоков тома. Остальные распечатки будут нам помогать.

Цифры нам пока мало о чем говорят. Придется использовать то немногое, что хоть

отчасти понятно. Например, что можно сказать о нулевом блоке?

В нем мы видим только одну членораздельную запись (в конце блока), ко-



торая начинается с адреса 750: «?BOOT—0—1/0 ERROR». Понятен только конец записи: ERROR означает «ошибка», 1/0 — типовое сокращение от слов INPUT/OUTPUT (ввод/вывод), так что это — «ошибка ввода/вывода». Ясно также, что это фатальная ошибка, после которой система не может завершить работу. Совершенно непонятно слово BOOT, в переводе с английского означающее «ботинок».

Чтобы разобраться с ним, возьмем «Словарь по компьютерам» под редакцией Ф. Гэлланда. В нем несколько слов начинаются с boot. Вот то, что нам нужно:

Boot — (1) операция загрузки операционной системы и, возможно, другого системного программного обеспечения (с магнитной ленты или диска).

Есть еще bootstrap; дословно это означает «ремешок ботинка» (русский аналог этого ремешка — «ушки» у сапог), но в компьютерном аспекте это начальная загрузка программы.

Обратим внимание, что Ф. Гэлланд различает boot (общая загрузка операционной системы или ее восстановление) и bootstrap (начальный ввод программы или операционной системы). Именно слово bootstrap породило весь набор понятий, связанных с загрузкой. Раньше на заднике ботинок делали ремешок в виде петли, чтобы удобнее было натягивать его на ногу, и в ситуации, когда мы вспоминаем Мюнхгаузена и говорим «поднять самого себя за волосы», англичане говорят «поднять самого себя за ремешки ботинок». Именно в такой ситуации оказывается ЭВМ, если ей надо загрузить программу, а программа загрузки программ в ОЗУ отсутствует. На выручку приходит «ремешок» — начальный загрузчик ОС.

Именно он записан в нулевом блоке.

Заметим: этот блок совпадает в основном с последним блоком драйвера DX.SYS, и вот почему: в нем записан не просто загрузчик системы, а загрузчик с устройства DX:

Зафиксируем наши новые знания.

0. Нулевой блок системного тома содержит начальный загрузчик системы, который копируется в него из последнего блока драйвера системного устройства.

Начиная со второго блока идут какие-то числа. Видимо, это текст программы. Кодировки команд мы не знаем, так что попытаемся найти что-нибудь понятное в колонке ASCII. Связные наборы символов (слова) встретятся нам в конце этой программы. То, что это сообщения загрузчика монитора, мы уже можем сказать точно. Приведем их здесь, переведем на русский язык и прокомментируем.

47

?BOOT—W—INVALID OR MISSING TT.SYS — неправильный или отсутствующий драйвер терминала.

Система работать будет, потому что драйвер системного терминала находится в резидентном мониторе, но при указании терминала как устройства в спецификации файла будет выдаваться сообщение, что такое устройство отсутствует.

?BOOT—W—SWAP.SYS NOT FOUND — файл SWAP.SYS не найден.

В этом случае для нормальной работы системы необходимо записать на системный том файл SWAP.SYS либо командами управления системой выбрать такой режим, в котором этот файл не нужен. Такие команды мы узнаем позже.

?BOOT—W—HANDLER FILE NOT FOUND — файл драйвера системного устройства не найден.

ОС RT-11 очень гибкая система, независимость ее модулей позволяет справиться чуть ли не с любой ситуацией. В данном случае загрузчик предупреждает, что ввиду отсутствия драйвера системного устройства монитор должен сам уметь работать с этим устройством (должен включать в себя соответствующий модуль или средство).

?BOOT—W—SWAP FILE TOO SMALL — файл SWAP.SYS мал.

Для нормальной работы можно предпринять такие же действия, как и при отсутствии файла SWAP.SYS. Однако если учесть, что он является буфером для хранения содержимого памяти при работе с





KMON и в других случаях, то достаточно просто увеличить его размер.

**?BOOT—W—ERROR READING HANDLER** — ошибка чтения драйвера.

Сообщение выдается при попытке прочитать драйвер устройства (для включения его в систему), если при этом происходит ошибка чтения из-за дефектов аппаратуры или носителя. Сообщение может выдаваться столько раз, сколько драйверов записано на системном томе, если он сильно «сбоит» (низкого качества). С системой работать можно, не пользуясь устройствами, драйверы которых не были прочтаны. В противном случае будет выдано сообщение об отсутствии данного устройства, несмотря на то что соответствующий драйвер (нечитающийся) есть на системном томе. К сожалению, загрузчик не печатает имя «плохого» драйвера.

**?BOOT—W—MONITOR FILE NOT FOUND** — файл монитора не найден.

Загрузчик должен загрузить в память монитор, имя которого записано в конце пятого блока. Не найдя в каталоге системного тома этот файл, загрузчик не знает, что делать дальше. Нужно помочь ему (записать на системный том нужный монитор) или «уволить» — записать другой загрузчик.

**?BOOT—W—CONFLICTING SYSGEN OPTION** — конфликт параметров генерации системы.

Это сообщение также следствие гибкости системы. При построении ОС RT-11 пользо-

ватель может указать некоторые параметры, в соответствии с которыми будут выбраны или изменены ее функции. Загрузчик монитора проверяет соответствие набора параметров монитора набору параметров драйверов. В случае несовпадения выдается это сообщение. Заметим, что некоторые драйверы, например SL.SYS, могут приспосабливаться к параметрам генерации системы.

А в конце пятого блока прямо указано, что это загрузчик монитора RT11SJ («BOTRT11SJ») с устройства DX: и записано приветствие системы «RT11SJ V05.00», которое мы видели на экране дисплея.

Но почему начальный загрузчик не загружает сразу монитор, зачем нужна промежуточная ступень — загрузчик монитора? По рассмотренным сообщениям видно, что, прежде чем передать управление монитору, его загрузчик проверяет аппаратуру и программы, соответствие параметров генерации и «докладывает» монитору реальное состояние программно-аппаратного комплекса.

Зафиксируем это.

2—5. Со 2-го по 5-й блоки записан загрузчик монитора, который кроме загрузки тестирует аппаратную и программную части системы и выполняет другие подготовительные операции, необходимые для работы монитора. Этот загрузчик переписывается специальными командами (о них разговор будет позднее) из 1—4-го блоков файла монитора.

А теперь займемся первым блоком. Сразу бросаются в глаза знакомые записи, которые мы видели при распечатке каталога тома.

С адреса 730 двенадцатью символами в коде ASCII записан идентификатор тома — RT11SJ V5.00.

С адреса 744 двенадцатью символами в коде ASCII записано имя владельца — АРХАНГЕЛСКИЙ.

И с адреса 760 также двенадцатью символами в коде ASCII записан текст, который мы раньше не видели, но сразу догадываемся, что это идентификатор системы: название фирмы-разработчика DEC и название операционной системы RT11A.

Здесь есть еще название версии операционной системы (в коде RADIX-50), которая инициировала (т. е. создавала вновь) файловую систему на томе — V05.

Теперь осталось только разобраться с двумя непонятными числами по адресам 722 и 724. Понятно, что весь первый блок тома — служебный и содержит необходимые для работы с томом признаки. Подумаем, какие именно могут потребоваться.

```

BLOCK NUMBER 00001
000/ 000000 000000 000000 000000 000000 000000 000000 000000 #.....*
020/ 000000 000000 000000 000000 000000 000000 000000 000000 #.....*
040/ 000000 000000 000000 000000 000000 000000 000000 000000 #.....*
060/ 000000 000000 000000 000000 000000 000000 000000 000000 #.....*
100/ 000000 000000 000000 000000 000000 000000 000000 000000 #.....*
120/ 000000 000000 000000 000000 000000 000000 000000 000000 #.....*
140/ 000000 000000 000000 000000 000000 000000 000000 000000 #.....*
160/ 000000 000000 000000 000000 000000 000000 000000 000000 #.....*
200/ 000000 000000 000000 000000 000000 000000 000000 000000 #.....*
220/ 000000 000000 000000 000000 000000 000000 000000 000000 #.....*
240/ 000000 000000 000000 000000 000000 000000 000000 000000 #.....*
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 000000 #.....*
720/ 000000 000001 000006 107123 052122 030461 045123 053040 #.....S.RT11SJ V*
      A      F      V05  MSZ  G43  K5*  M1H
740/ 027063 030060 071141 060550 063556 066143 065563 065151 #5.00АРХАНГЕЛСКИЙ*
      BD/   G.P  RLA  OWP  PUB  QM/  QG*  QAA
760/ 042504 051103 030524 040461 020040 020040 000000 000000 #DECRT11A  ....*
      KCL  MFS  G5.  J01  EEX  EEX

```

Система должна иметь возможность найти на томе файлы, записи о которых хранятся в каталоге. А где находится каталог? Он начинается с 6-го блока. Это мы видим по

дампу; это же число записано по адресу 724. Размер каталога нам сообщила программа DIR — 1 сегмент; это число мы видим по адресу 722.

BLOCK NUMBER 00006							
	A		A		H		
000/	000001	000000	000001	000000	000010		
012/	102000	075131	062000	075273	000032	000000	002654
	UD2	SWA	F	SYS	Z		6L
030/	102000	071677	142302	075273	000111	000000	006514
	UD2	RT1	1SJ	SYS	A3		BED
046/	102000	100040	000000	075273	000002	000000	006514
	UD2	TT		SYS	B		BED
064/	102000	016300	000000	075273	000003	000000	006514
	UD2	DX		SYS	C		BED
102/	102000	074240	000000	075273	000015	000000	006514
	UD2	SL		SYS	H		RED
120/	102000	046600	000000	075273	000002	000000	006514
	UD2	LP		SYS	H		BED
136/	102000	052400	000000	075273	000012	000000	006514
	UD2	HX		SYS	J		BED
154/	102000	054540	000000	075273	000002	000000	006514
	UD2	NL		SYS	B		BED
172/	102000	071070	000000	075273	000003	000000	006514
	UD2	RK		SYS	C		BED
210/	102000	052140	000000	075273	000011	000000	006514
	UD2	MT		SYS	I		BED
226/	102000	015600	000000	075273	000003	000000	006514
	UD2	DP		SYS	C		BED
244/	102000	045640	000000	075273	000010	000000	006514
	UD2	LD		SYS	H		BED
262/	002000	062570	000000	073376	000035	000000	004314
	YX	FIP		SAV	/		APL
300/	002000	016130	000000	073376	000055	000000	004314
	YX	DUP		SAV	AE		APL
316/	002000	015172	000000	073376	000023	000000	004314
	YX	DIR		SAV	S		APL
334/	002000	016125	062000	073376	000011	000000	004314
	YX	DUM	P	SAV	I		APL
352/	002000	016336	023364	015172	000002	000001	000000
	YX	DX0	FIL	DIR	B	A	
370/	002000	016306	035045	015172	000002	000000	017656
	YX	DXF	ILE	DIR	B		ERO
406/	001000	007347	012170	015622	000342	000000	006156
	L2	B00	CK	DPR	EZ		A9V
424/	004000	007347	012170	015622	000161	000000	006156
	AKN	B00	CK	DPR	B3		A9V
442/	004000	026676	140700	015533	000160	000000	031655
	AKN	GLO	1	DOC	F2		HJ/
460/	004000	026676	140700	015533	000160	000000	031655
	AKN	GLO	1	DOC	B2		HJ/
476/	004000	016125	062000	073376	000335	000000	026715
	AKN	DUM	P	SAV	EU		GME
514/	004000	000000	000000	000000	000000	000000	000000
		AKN					

Зафиксируем:

1. Первый блок содержит идентификаторы тома, владельца и системы, а также справочную информацию для файловой системы.

Итак, мы познакомились с загрузчиками — начальным, «живущим» в нулевом блоке, и загрузчиком монитора, «живущим» в блоках 2—5. В процессе своей работы они тестируют программы и аппаратуру. В младших версиях RT-11 загрузчики проводили меньше тестов, поэтому при загрузке разных версий системы на одной и той же аппаратуре получались неодинаковые результаты. Так, младшие версии могли загрузить монитор на частично неисправной

аппаратуре и отказаться работать на исправной. Например, загрузчик V3.0 (и младше) проверял функционирование и размер оперативной памяти процессора только путем чтения. Если же в ее составе оказывалось ПЗУ, то, безуспешно пытаясь на следующем шаге работы записать туда информацию, он в итоге прекращал работу, не выдавая никаких сообщений.

До V4.0 не проверялись драйверы устройств (и соответственно не включались в систему). Это должен был делать пользователь.

Так как первичный загрузчик записывается из последнего блока драйвера системного устройства, то при выдаче сообщения

?BOOT—U—I/O ERROR первое, что следует сделать,— записать новый драйвер системного устройства и снова переписать загрузчик.

Однако может испортиться и вторичный загрузчик; так как он переписывается из первых блоков монитора, то второй путь восстановления загрузчика — заменить файл монитора.

Но самое обидное — когда у тома, заполненного информацией, вдруг портится первый (системный) блок. В таких случаях программа DIR выдает соответствующее сообщение и отказывается работать. Сразу предупредим, что и тогда можно спасти файлы, хотя действовать придется вслепую. Но об этом потом, разберемся сначала с «домовой книгой».

### 6.3. Заглянем в домовую книгу, или Структура каталога

Продолжая просматривать дампы, в 6-м блоке мы видели информацию, знакомую нам по работе с программой DIR,— имена и типы файлов, которые находятся на томе. Понятно, что это и есть наша домовая книга — каталог тома.

Для его анализа нам потребуется полная распечатка каталога в восьмеричной системе счисления в одну колонку. Кроме того, заметив периодичность чисел в 6-м блоке, перепишем дампы в другом виде.

Сравнив полученную таблицу с каталогом, мы видим, что во всех ее строках (кроме первой) записана информация о файлах. Разобьем строку на 16-разрядные слова.

Нулевое слово содержит какой-то признак (или сумму признаков). Так, записи с содержимым нулевого слова, равным 102 000, соответствуют файлам (в каталоге) с признаком P. В то же время все записи, соответствующие файлам, которые мы видим в распечатке каталога, содержат в нулевом слове число 2000. Кроме того, есть записи о файлах, которые в распечатку каталога не попали; там в нулевом слове стоят числа 1000 или 4000. Будем надеяться, что в дальнейшем удастся выяснить, что есть что.

Первое, второе и третье слова записей, как видно невооруженным глазом, содержат имена и типы файлов в коде RADIX-50. Но вот в следующем, четвертом, слове содержится не столь понятное число. Если мы поищем эти числа в распечатке каталога, то увидим, что это размер файла в восьмеричной системе счисления. Пятое слово содержит нуль, в шестом опять какая-то константа. В распечатке каталога остался

только один признак, который мы еще не связали с записью,— дата, так что можно предположить, что шестое слово содержит дату.

Вернемся к нулевому слову. Сравним «по-файлово» записи в дампе и каталоге и увидим:

все записи о файлах, которые мы видим в распечатке каталога, имеют в нулевом слове константу 2000;

все записи о файлах, у которых в распечатке каталога стоит символ P, имеют в нулевом слове константу 102 000, которую можно представить как сумму  $100\ 000 + 2000$ ;

запись, которая в распечатке каталога обозначается словом UNUSED, в дампе отображается как запись о файле BOOCK.DPR с размером, равным размеру неиспользуемой зоны, и константой 1000 в нулевом слове;

просуммировав размеры файлов, перечисленных в дампе до первой записи с константой 4000, мы видим, что они занимают том полностью. Файлы, записи о которых начинаются с константы 4000, в распечатке каталога отсутствуют. Можно предположить, что константа 4000 соответствует концу каталога.

Для удобства дальнейшей работы разделим нулевое слово на два байта и выпишем встретившиеся варианты. Получилась интересная таблица:

001 000=002 000	— неиспользуемый
002 000=004 000	— постоянный
004 000=010 000	— маркер конца каталога
100 000=200 000	— защищенный

Видно, что используется только один байт. Так как признаки состояния возрастают монотонно и один признак, о котором мы говорили, отсутствует («временный»), то, очевидно, ему должен соответствовать код 001 (или 000 400, если записать его в виде слова).

Кстати, когда мы распечатывали суммарную информацию о каталоге, говорилось о том, что наш каталог состоит из одного сегмента. Откроем маленькую тайну — код 010 означает маркер конца сегмента, а не каталога.

Теперь можно уточнить таблицу признаков состояния файла, содержащихся в нулевом слове записи о файле в каталоге.

000 400=001 000	— временный
001 000=002 000	— неиспользуемый
002 000=004 000	— постоянный
004 000=010 000	— маркер конца сегмента
100 000=200 000	— защищенный

Особо поговорим о временных файлах. Запись о них в распечатке каталога мы можем никогда не увидеть, потому что со-



стояние «временный» у файла появляется только во время работы какой-либо программы при проведении операций вывода информации в файл. Пользователь может столкнуться с этим состоянием при аварийном завершении программы (сбой по питанию, сбой аппаратуры, ошибка в программе). В такой ситуации важно сохранить хотя бы ту информацию, которая уже записана в файл. Для этого в системе есть специальная команда CLOSE (закрыть). Синтаксиса она не имеет, и если набрать .CLOSE, то система закроет все открытые файлы, т. е. изменит признак «временный» на «постоянный».

Даже если информация не нужна, системе необходимо привести в устойчивое состояние. Для этого следует освободить открытые файлы командой PURGE (освободить), которая также не имеет синтаксиса и действует на все открытые файлы. Она изменит у них признак «временный» на «неиспользуемый».

А что такое файлы, имеющие признак 4000? Они существуют, но в «потусторон-

нем мире». Система при инициализации нового каталога не стирает информацию о старых файлах, а записывает в нулевое слово первой записи каталога о файле маркер конца сегмента. Это полезно вспомнить, когда возникает необходимость восстановить случайно удаленную информацию.

Разберемся, наконец, с последним словом записи о файле. Мы догадались, что оно содержит дату, но в каком коде (или в каком формате) она записана? Для решения этой головоломки выпишем в отдельную таблицу коды из шестых слов записей и соответствующие им даты из распечатки каталога:

002 654=13—JAN—84  
 006 514=10—MAR—84  
 004 314=06—FEB—84  
 017 656=21—JUL—86  
 006 156= . . . . .

Ни одним из известных нам кодов невозможно записать дату в одном слове. По видимому, каждый элемент даты закодирован каким-то двоичным числом, а потом из этих чисел составлено 16-разрядное слово. Посмотрим, как это можно сделать.

В месяце может быть 31 день. Для записи кода дня необходимо пятиразрядное двоичное число, так как из пяти двухзначных цифр можно составить 32 комбинации.

Год состоит из 12 месяцев, следовательно, для кодирования месяца необходимо четырехразрядное двоичное число.

А как же год? Из 16 разрядов слова остались незадействованными только 7; из них можно составить 128 комбинаций. Поэтому весьма вероятно, что закодированы только последние две цифры года.

Дальнейшая наша работа будет мало отличаться от работы дешифровальщика. Перепишем предыдущую таблицу, добавив запись зашифрованной даты в двоичном коде и двоичные коды дня и месяца.

Дата	Код даты	Двоичный код даты	Двоичный код месяца	Двоичный код дня
13—JAN—84	002654	0.000.010.110.101.100	0001	01101
06—FEB—84	004314	0.000.100.011.001.100	0010	00110
10—MAR—84	006514	0.000.110.101.001.100	0011	01010
21—JUL—86	017656	0.001.111.110.101.110	0111	10110

Попробуем двоичный код месяца в первой строке подставить в двоичный код даты так, чтобы значения их разрядов совпали:

0.000.010.110.101.100  
 00.01

Как видим, это можно сделать единственным способом. Но не будем спешить радоваться. Для проверки проделаем такую же операцию с остальными строками, подставляя код меся-

ца в уже найденное место.

0.000.010.110.101.100  
 00.01  
 0.000.100.011.001.100  
 00.10  
 0.000.110.101.001.100  
 00.11  
 0.001.111.110.101.110  
 01.11

Проверка показала, что место месяца в коде даты мы определили правильно. Во второй строке мы могли сделать подстановку в другое место, но при таком варианте были бы неправильны подстановки в других строках.

Теперь повторим все, что мы делали с кодом месяца, но уже с кодом дня.

0.000.010.110.101.100

0.110.1

0.000.100.011.001.100

0.011.0

0.000.110.101.001.100

0.101.0

0.001.111.110.101.110

1.010.1

Мы смогли выделить группу разрядов, соответствующих кодам дня, но на код года остается только пять разрядов! Расшифруем их.

Год 1984=01.100=12.

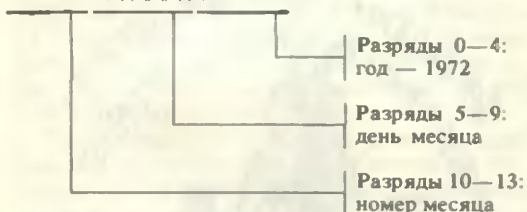
Год 1986=01.110=14.

Совсем не то, что нам нужно. Однако обратим внимание на то, что разница между кодом года и действительным значением года одинакова и равна 1972. Пожалуй, самое время вспомнить, что разработка RT-11 была начата в 1972 г.

Построим сводную таблицу.

0.000.000.000.000.000

ММ ММД ДДД ДГГ ГГГ



Используя ее, определим дату создания файла, находящегося на месте, которое в распечатке каталога показано как неиспользуемое. По дампу видно, что это

BOOSK.DPR 226 03—MAR—86 286

Действительно, именно так выглядит запись в каталоге удаленных файлов.

Исследование остальных файлов, которые оставили свои следы в 6-м блоке, мы предоставляем читателю в качестве упражнения.

#### 6.4. А что же на обложке, или Заголовок сегмента каталога

Нам осталось разобраться с первой записью в дампе 6-го блока (она называется заголовком сегмента каталога), но прежде посмотрим, как можно вычислить номер блока, с которого начинается какой-нибудь файл.



В записи о файле есть только один атрибут его физического представления на носителе — размер файла. Но так как файлы записаны подряд и сразу после конца одного следует начало другого, то адрес файла можно найти как сумму размеров всех предшествующих (в сегменте) файлов. Необходимо только знать номер блока, с которого начинается самый первый файл, записанный в сегменте. Этот номер и записан в четвертом слове заголовка сегмента каталога.

Отметим, что неиспользуемая область на томе — это также файл с именем EMPTY.FIL, если здесь ничего не было в прошлом, или с именем файла пользователя, который был на этом месте.

А сколько записей о файлах может быть в сегменте? Подсчитаем.

Сегмент каталога состоит из 512. слов, 5 из которых занимает заголовок сегмента; запись о файле состоит из 7 слов; сегмент каталога может содержать

$$N = \frac{512 - 5}{7} = 72. \text{ записи.}$$

Однако маркер конца сегмента имеет такой же формат, что и запись о файле. Кроме того, сегмент должен заканчиваться записью о неиспользуемой области, хотя бы и имеющей нулевой размер. Следовательно, в одном сегменте можно описать 70 файлов.

Обратите внимание: размер устройства DX: — 486 свободных блока, и если записать на него файлы размером по одному блоку, то придется завести каталог размером 432 записи, что больше одного сегмента. Но сколько же сегментов потребуется в таком случае — 7 или больше?

А вот еще один вопрос. Пусть у нас есть 200 файлов по 2 блока, и взяли мы для каталога 3 сегмента, и записывали файлы подряд, без промежутков. А потом понадобилось нам удалить файл, описание которого хранится в первом сегменте, и вместо него записать два файла по одному блоку. Но место-то для записи освободилось только одно! Как быть?

Разработчики фирмы DEC решили полностью заполнять только последний сегмент каталога, а предыдущие — лишь наполовину. При таком подходе между двумя записями о файлах можно расположить еще одну, например о неиспользуемом файле. Но из-за этого в каждом сегменте (кроме последнего) может быть не более 36 записей о постоянных файлах.

При заполнении последнего сегмента открывается следующий, и половина записей из предыдущего переписывается в него. Значит, для 200 файлов потребуется не 3, а 5 сегментов. Единственный случай, когда все сегменты, имеющие записи, заполняются полностью, возникает при «сборке мусора», когда мы просим систему собрать все свободные области в одну, а заполненное файлами пространство сжать в начало тома.

Отсюда следуют некоторые выводы. Если мы хотим зарезервировать для каталога 10 сегментов, то должны сообщить об этом системе. Для этого служат нулевые слова заголовков сегментов каталога, в которые записывается число зарезервированных сегментов. Если же записи о файлах у нас находятся, например, только в 5 сегментах, то (чтобы система не тратила время на просмотр пустых сегментов) нужно указать, сколько сегментов открыто. Для этого используется второе слово заголовка первого сегмента каталога.

А теперь представим, что в результате всякого рода перипетий сегмент номер 3 освободился, а потом открылся снова. Поэтому и записи о файлах расположились не так, как сами файлы на устройстве. Связи между сегментами при этом выглядят, как на рис. 1. Следовательно, необходимо в каждом сегменте указать, какой сегмент

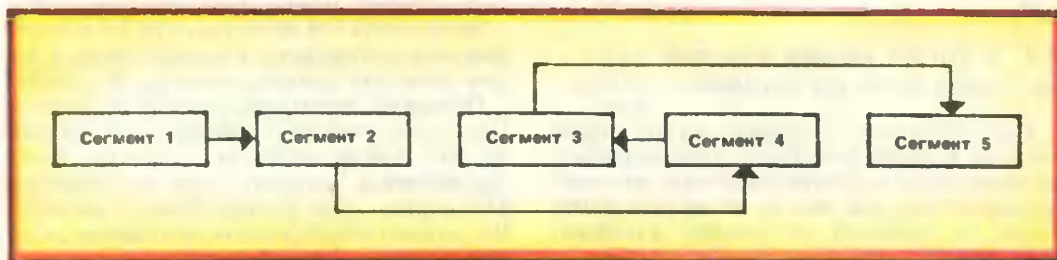
просматривать следующим. Поэтому в первое слово заголовка каждого сегмента каталога записывается номер следующего сегмента. В последнем используемом сегменте первое слово содержит нуль.

И последнее. У нас есть возможность поместить в запись о файле какие-нибудь свои признаки. Тогда запись будет состоять не из 7, а из большего числа слов. Система, правда, не будет обрабатывать дополнительные слова, нам придется позаботиться о такой обработке самим. Но сообщить файловой системе о числе дополнительных слов в записи необходимо, и записывается эта информация в третье слово заголовка сегмента. Число дополнительных слов может быть определено при инициализации каталога.

А теперь просуммируем все, что у нас получилось.

Нулевое слово сегмента каталога хранит число сегментов, отведенных под каталог. Число сегментов задается по умолчанию (разное для различных устройств, в зависимости от объема тома) или по указанию пользователя при инициализации каталога. В нашем случае оно равно 1.

Первое слово хранит номер следующего сегмента каталога. Оно является связующим между логически смежными сегмен-





тами. В последнем используемом сегменте в первом слове записан нуль.

Второе слово — это счетчик сегментов, имеющих записи. При открытии нового сегмента счетчик корректируется только в первом сегменте. В последующих сегментах содержимое этого слова не изменяется и равно нулю.

Третье слово — число дополнительных слов в записи о файле в каталоге.

В четвертом слове хранится номер блока, с которого начинаются файлы, описанные в данном сегменте. В нашем случае он равен 10.

### 6.5. Структура тома с произвольным доступом

Наше «расследование» подошло к концу. Полученная информация не раз пригодится в работе, и для лучшего запоминания сделаем «дайджест» характеристик тома с произвольным доступом.

Заголовок тома расположен в нулевом блоке и начинается с кода 000 240. В случае отсутствия этого кода аппаратный загрузчик будет пытаться прочитать этот том снова. В заголовке тома расположен первичный загрузчик, который «знает», где находится основной загрузчик системы; вся его задача сводится к загрузке основного загрузчика. Если том не системный и загрузка как операция не определена, то в нулевом блоке вместо загрузчика записана программа вывода на терминал сообщения BOOT NO VOLUME — «загрузчика нет на томе».

Идентификатор тома расположен в первом блоке и содержит: число сегментов каталога; номер блока, с которого начинается каталог; название версии системы, которая инициализировала каталог, в коде RADIX-50; идентификатор тома и имя владельца, введенные оператором при инициализации каталога; название фирмы-разработчика и название операционной системы.

Основной загрузчик монитора расположен в блоках со 2-го по 5-й и служит для первоначальной (или повторной) загрузки операционной системы.

Каталог тома расположен с 6-го блока тома, состоит из сегментов, имеющих размер 2 блока. Число сегментов, отведенных под каталог, указано в нулевом слове каждого сегмента.

Все разговоры о расположении каталога и файлов на томе нам удалось провести, не упоминая об устройстве, на котором находится том. С точки зрения файловой системы RT-11 том есть некоторая область памяти, имеющая стандартный для системы

заголовок тома и каталог. Том можно организовать на любой памяти. Например, если в начало файла на устройстве с произвольным доступом мы запишем каким-нибудь образом заголовок тома и каталог, то сможем написать и программу, которая работает с нашим файлом как с томом. Программы, позволяющие создавать такие файлы-тома и работать с ними, появились в третьей версии RT-11 (драйверы UX, UY, UZ, работающие с файлами различного размера — 280, 560 и 1100 блоков соответственно). Ну а если мы создали устройство UZ: размером 1100 блоков, то имеем право создать на нем файл размером 560 блоков, объявить его устройством UY: и работать с ним как с самостоятельным устройством. Внутри устройства UY: мы можем, как в матрешке, создать еще и устройство UX:. Как и все устройства в RT-11, создаваемые нами логические устройства-файлы могут иметь номера от 0 до 7.

Но работать с файлами фиксированных размеров не всегда удобно. Поэтому в версии RT-11 V4.0 для устройств UX:, UY:, UZ: размеры файлов стало можно определять любые.

И наконец, в версии V5.0 драйверы UX, UY, UZ, а также программы, работающие с этими устройствами, были исключены. Организация логических дисков (logical disk = LD:), как они стали называться, — инициализация каталога, связывание имени файла с именем устройства и его номером — была возложена на программу DUP, а для работы с файлом в систему был включен драйвер LD.SYS.

Таким образом, теперь у нас есть устройства LD: с номерами от 0 до 7, которые можно вкладывать друг в друга, как матрешки. Это оказалось очень удобным. Расходы па-

55



мяти на организацию каталога окупаются более быстрым обращением к нужному файлу, так как каталог становится короче.

В качестве примера рассмотрим рис. 2. Про изображенный на нем том можно сказать следующее:

он находится на устройстве RK0: и содержит два больших файла — WORKIN.DSK и SOURCE.DSK;

файл SOURCE.DSK в свою очередь содержит заголовки тома LD2, каталог и файлы PG.MAC, PG.PAS, PG.TXT;

файл WORKIN.DSK содержит заголовок тома LD1, каталог и файлы PRG.MAC и MACRO.DSK;

файл MACRO.DSK содержит заголовок тома LD3, каталог и файлы P1.MAC и P2.MAC.

Таким образом, чтобы добраться до файла P2.MAC, мы должны идти таким путем: устройством LD1: считать WORKIN.DSK на RK0;

56 устройством LD3: считать MACRO.DSK на LD1:.

Теперь можно сказать, что файл P2.MAC находится на устройстве LD3:, и работать с ним.

Итак, каждый файл может быть смонтирован (есть такое программистское выражение) на устройство LD: с любым номером. Мы получили возможность производить операции с группой файлов, рассматривая их как один. Это особенно важно при работе с магнитной лентой.

А теперь вспомним: в «сотке» есть дополнительная память у процессора. В ней тоже можно организовать подобное устройство! В RT-11 V5.0 для этого есть устройство VM: (virtual memory — виртуальная память) и соответствующий драйвер. Физически эта та же быстрая память процессора, и при выключении питания информация в ней пропадает. Но логически это устройство с произвольным доступом. Так как память у процессора одна, то номер у устройства VM: игнорируется. Еще одно отличие от устройства LD: в том, что устройство VM: может быть системным, т. е. в заголовке тома можно записать загрузчик монитора и загрузить систему с VM:. Что этим достигается? Чрезвычайно высокая скорость работы!

Если дополнительной памяти мало (такова, увы, «сотка»), то устройство VM:

используется в основном как буферное для различных программ. А вот у SM-1420, «Электроника-60.1», «Электроника-79», «Электроника-85», ДВК-3 объем памяти может достигать 4М байт, и тогда на VM: можно перенести всю операционную систему.

### 6.6. Как лиса сыр делила, или Дисциплина распределения памяти

Тома могут быть большими и маленькими, содержать один файл или сотню. И всегда для работы нужна свободная область (CO) на томе, куда будет записан очередной файл. А если свободных областей много и все они разных размеров, в которую из них система определит файл? Это может нас интересовать, когда неизвестно, хватит ли места для очередного файла или, что более важно, где искать информацию, если произошла какая-нибудь авария.

Начнем с самого простого случая — мы знаем (или догадываемся), каким будет размер выходного файла. При выборе места для него тактика очевидна — следует использовать наименьшую из тех CO, в которые он поместится. Именно так и поступает RT-11, если размер образуемого файла мы сообщили ей при помощи ключа /ALLOCATE.

Более сложен случай, когда мы (а система тем более) не знаем размера выходного файла. Кроме того, следует помнить, что одновременно могут быть открыты три (иногда и более) выходных файла, а так как открываются они по очереди, мы должны учесть и это.

Первое, что приходит в голову, — выбирать всегда самую большую CO. Вроде бы надежно: если уж файл не поместится в нее, то его вообще нельзя записать на том. А если мы открываем два файла, причем второй будет больше первого? Тогда второй файл может и не поместиться, хотя его размер меньше самой большой CO.

Проявим благоразумие и позаботимся о завтрашнем дне. Будем всегда брать вторую по величине CO. Но при этом мы впадаем в другую крайность: оба файла могут не поместиться, хотя имеется огромная, самая большая CO.

Разработчики RT-11 приняли остроумное решение. Не рассматривая самые маленькие CO, они делают выбор из самой



большой и второй по размеру СО. Такая дисциплина (или алгоритм) распределения памяти преследует простую цель — обеспечить двум последовательно открытым файлам примерно одинаковые условия. Реализуется она так.

1. Если вторая по размеру СО больше половины первой, ее получает первый файл; второй файл получает половину первой СО.

2. Если первая по размеру СО более чем вдвое превосходит вторую, то первый файл получает половину первой СО; второй файл получает большую из следующих областей: вторая по размеру СО и четверть первой СО.

В любом случае для двух файлов разница в размерах не будет превышать половины самой большой СО.

На рис. 3—5 проиллюстрированы случаи, когда вторая СО оказывается примерно рав-



5

на первой, много меньше ее и много меньше ее половины.

И в какой же половине наибольшей СО нам искать информацию в случае нужды? При какой-либо аварии, когда не установлен признак «постоянный» выходным файлам, программа DIR на месте разделенной СО покажет две области, одинаковые по размеру (или отличающиеся на один блок) и следующие друг за другом. В первой из них (от начала тома) и будет находиться наша выходная информация.

57

Есть еще один способ. При открытии файла прикладная программа дала ему имя и тип. Если вывести на дисплей каталог удаленных файлов, то мы увидим запись о файле с этим именем, и, соответственно, на его месте следует искать информацию.

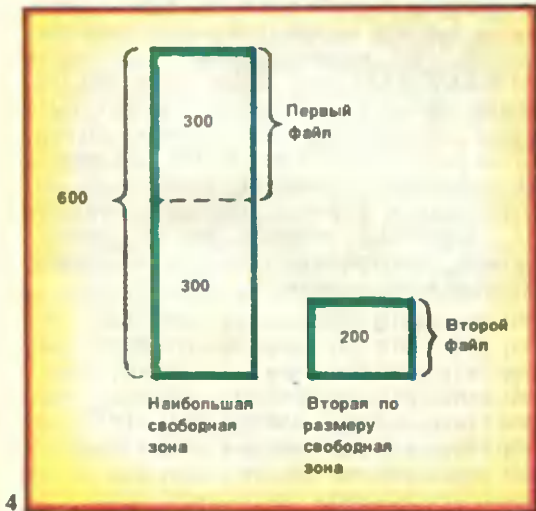
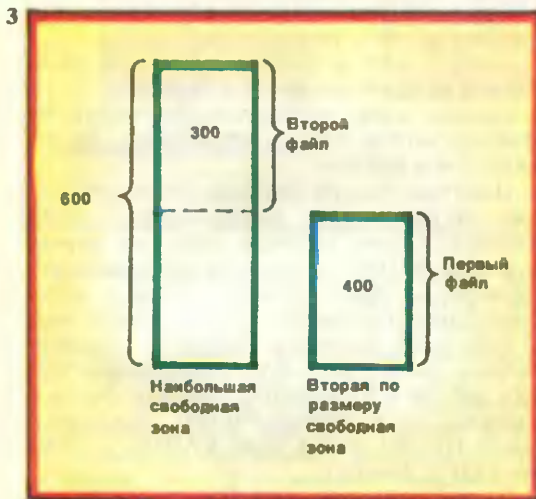
Остается только восстановить запись в каталоге.

### 6.7. Для чего это нужно, или Восстановление доступа к информации

Способ поиска информации, рассмотренный при анализе дисциплины распределения памяти, позволяет восстановить доступ к информации, используя данные в каталоге тома. А как быть, если каталог тома испорчен? Как узнать, где файл начинается, где кончается и какая информация в нем записана?

Нас выручит программа DUMP.

Для начала разделим информацию: что относится к одному файлу, что — к другому. Для этого нужно примерно определить назначение файла, тип информации в нем. Дамп любого файла имеет свой характерный «ритм» чисел, отображающих информацию. По изменениям этого «ритма»



4

мы и будем находить конец и начало файла.

Файл DXFILE.DIR мы создали сами с помощью программы DIR; он текстовый. В его дампе мы действительно увидим, что символы, получающиеся из двоичных кодов, составляют текст, который начинается с начала нулевого блока файла. Последний же его блок частично заполнен текстом, а оставшаяся часть заполнена нулями. Таким образом, если несколько текстовых файлов расположены друг за другом, то граница между ними легко находится по нулевым байтам в конце последнего блока предшествующего файла и осмысленному тексту с начала нулевого блока последующего файла. Остается только занести полученные данные в восстанавливаемый каталог.

А как выглядят программы в файлах, имеющих тип .SAV? Например, дампы программы DUMP?

58

Он начинается с нулей, и только 3-я строка с адреса 000 040 заполнена какими-то числами. Часть чисел имеет значение только для системы и программистов, поэтому мы их рассматривать пока не будем. Сейчас нас интересуют лишь адреса начала программы и ее конца. Впрочем, начало программы мы уже увидели — это блок, начинающийся с нулей и с характерно заполненными 3-й и 16-й строками.

Конец программы должен быть заведомо дальше верхнего адреса\* программы, который нужно искать в ячейке по адресу 000 050. В нашем случае это значение равно 10 134 (восьмеричное). И размер файла можно определить, отбросив последние три цифры этого числа и прибавив к оставшейся части единицу.

Теперь мы знаем всё, что нужно для восстановления записи в каталоге, — номер начального блока и размер файла.

Проверим себя и посмотрим дампы программы DUP.SAV. Всё вроде бы хорошо — и начальный блок нашли правильно, и числа в общем-то похожи. Но вычисленный размер файла оказался меньше действительного. В чем дело?

Поищем отличия в нулевом блоке от аналогичной информации в программе DUMP. Чисел в нем больше! Например, в ячейке 44 вместо 000 000 записано 021 000 (в других подобных программах — 001 000). Какой-нибудь признак? Да. Это число (001 000) говорит о том, что размер файла с программой больше, чем число ячеек, занимаемых программой в ОЗУ.

Такие программы называют оверлейными или программами с перекрытиями. Их особенность в том, что совсем не обязательно хранить в ОЗУ всю программу; можно загрузить только основную часть (корневой сегмент), а фрагменты, реализующие отдельные функции (оверлейные сегменты), загружать на одно и то же место в памяти по мере необходимости. На рис. 6 изображено, как выглядят файл программы и программа, загруженная в ОЗУ.

Дополнительные числа в нулевом блоке нужны, чтобы система знала, как выделять память для программы.

Размер программы, как и раньше, записан в ячейке 000 050; как и раньше, он определяет размер ОЗУ, необходимый для программы, но равен здесь не полной ее длине, а сумме размеров корневого сегмента и самого большого оверлейного сегмента.

Системе необходимо также знать:

адрес памяти, куда каждый оверлейный сегмент должен загружаться;

номер блока в файле, с которого начинается каждый оверлейный сегмент;

сколько слов необходимо прочитать из файла, чтобы весь оверлейный сегмент оказался в памяти.

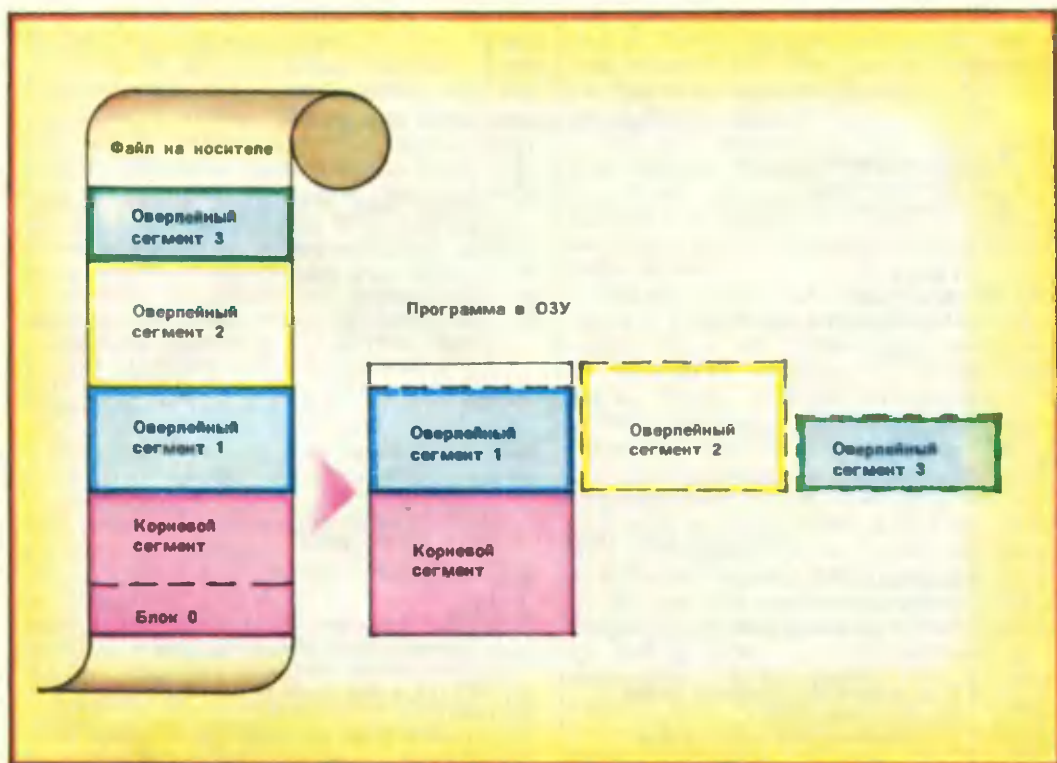
Понятно, что при большом числе оверлейных сегментов эту информацию удобнее хранить в виде таблицы; адрес ее указан в ячейке 000 064. Этот адрес указывает, как правило, на область первого блока, к которому стоит присмотреться повнимательнее.

Если наша программа действительно оверлейная, имеет тип .SAV и предназначена для работы в SJ-мониторе или как фоновое задание, то в ячейке 001 000 содержится число 060 502, что в коде RADIX-50 означает OVR (overly).

Кроме того, как уже было сказано, по указанному в ячейке 000 064 адресу находится таблица параметров оверлейных сегментов. В нашем случае (программа DUP.SAV V5.1) этот адрес равен 001 106. Набор чисел в этой таблице может быть узнан по характерному «ритму». Первое слово указывает на адрес в ОЗУ для загрузки оверлейного сегмента, второе — на номер блока в файле, с которого начинается оверлейный сегмент; третье слово — размер оверлейного сегмента в словах. Учитывая это, перепишем таблицу:

001 106)	010 560	000 011	002 354
001 114)	010 560	000 016	001 565
001 122)	010 560	000 022	001 164
001 130)	010 560	000 025	002 144
001 136)	010 560	000 032	001 617
001 144)	010 560	000 036	002 430
001 152)	010 560	000 044	001 355
001 160)	010 560	000 047	002 420

\* На фирме DEK принято ОЗУ изображать вертикально; номера ячеек нарастают снизу вверх.



Ее конец можно найти опять же по нарушению «ритма». Следом идет другая таблица, которой мы до поры до времени заниматься не будем, и ее строки выглядят так: 004 357, 001 002, 000XXX, 0XXXXX.

А по параметрам последнего оверлейного сегмента можно определить размер файла (начало мы уже нашли). В нашем случае начальный номер последнего оверлейного сегмента равен 000 047, его размер — 002 420 слов (5040 байт). Так как он занимает неполные 6 блоков, то размер файла равен  $47 + 6 = 55$  блоков. Распечатка каталога показывает, что мы не ошиблись.

Есть и другие виды программ, но в них тип указан уже в нулевом блоке в коде RADIX-50.

Например, оверлейные сегменты можно хранить не на носителе, а в дополнительной памяти процессора. Такие программы называются виртуальными и в нулевой ячейке содержат признак VIR (в коде RADIX-50 это 105 372); в остальном они аналогичны оверлейным программам.

Есть программы, которые можно загружать в любые адреса памяти. Они имеют тип .REL (relocation — перемещать), могут быть как обычными, так и оверлейными. В них есть таблица перемещений, по которой система сама «доделывает» про-

граммы перед загрузкой в память. При работе с ними, вычислив обычным образом размер файла, мы должны еще найти конец таблицы перемещений. Это просто — ни один из кодов в таблице не равен —2 (или 177 776); именно это число выбрано в качестве признака конца таблицы. Разность между номером следующего блока и номером блока начала файла даст размер файла. Признак REL в таких программах записан в слове 000 060, а в следующем слове хранится номер блока, с которого начинается таблица перемещений.

### 6.8. Если быть последовательными, или

#### Файлы на магнитной ленте

До сих пор мы рассматривали устройства файловой структуры с произвольным доступом. Есть еще группа устройств, при работе с которыми неудобно писать или читать информацию из произвольных мест носителя, хотя бы потому, что очень велико время поиска этого места. Как правило, это накопители информации на магнитной ленте.

Они входят в группу устройств файловой структуры лишь потому, что определение имени и типа файла как метки, по которой его можно найти, имеет смысл и для них.

№ байта	Имя поля	Длина поля	Содержание поля
<i>Volume header label (VOL1) — метка начала тома</i>			
1—3	Идентификатор метки	3	
4	Номер метки	1	
5—10	Идентификатор тома	6	
11	Связка	1	Метка тома. Если вы не указали идентификатор тома во время инициализации, то по умолчанию RT11A Код пробела (32.)
12—37	Зарезервировано	26	
38—50	Идентификатор владельца	13	
			38-й байт-D } Это значения для лент, 39-й байт-% } записанных фирмой 40-й байт-B } DEC, (PDP-11) Байты 41—50 — имя владельца (максимум 10 символов) (по умолчанию — код пробела)
51	Стандарт версии DEC	1	
52—79	Зарезервировано	28	
80	Стандартная версия метки	1	3
<i>File header label (HDR1) — метка начала файла</i>			
1—3	Идентификатор метки	3	
4	Номер метки	1	
5—21	Идентификатор файла	17	
22—27	Идентификатор установки файла	6	6 символов имени файла в коде ASCII, точка 3 символа типа файла (имя файла до 6 символов можно дополнить кодом пробела) RT11A и код пробела
28—31	Номер секции файла	4	
32—35	Последовательный номер файла	4	Первый файл на ленте имеет номер 0001. Значение номера увеличивается на 1 для каждого последующего файла. В только что инициализированной ленте это значение равно 0000
36—39	Номер генерации	4	
40—41	Версия генерации	2	0001 00
42—47	Дата создания	6	
			Код пробела, затем последние две цифры года, умноженные на 1000, плюс номер дня (в коде ASCII); код пробела, затем 0000, если даты нет. Пример: 1/II—75 запоминается, как 75032
48—53	Срок хранения	6	
54	Доступность	1	Код пробела и 00000 указывают на истекший срок хранения файла
55—60	Счетчик блоков	6	
61—73	Код системы	13	DECRT11A
74—80	Зарезервировано	7	
<i>First end of file label (EOF1) — метка конца файла, полностью повторяет метку HDR1, за исключением: в байтах 1—3 записаны символы EOF, в байтах 55—60 записан номер последнего блока данных начиная от метки HDR1.</i>			

В остальном же файловая структура магнитных лент весьма своеобразна и отлична от принятой в RT-11 для дисков.

Наиболее распространенный носитель — катушечная магнитная лента (МЛ) — используется на различных магнитофонах. Для работы с ней в RT-11 используется файловая структура, определенная в стан-

дарте ANSI X3.27 уровень 1 (ANSI — American National Standards Institute — Американский национальный институт стандартов). Главное отличие от файловой структуры дисков — отсутствие каталога как такового. Вместо него используются метки файлов, в которых записана некоторая информация о файле, а также метка

тома, содержащая (подобно 1-му блоку на диске) информацию о томе.

То, о чем мы так долго говорили, — где файл начинается и где кончается — на МЛ описывается прямо с помощью метки начала файла (HDR1) и метки конца файла (EOF1). А чтобы данные нельзя было спутать с метками, последние заключаются как бы в скобки специальными кодами, которые принято называть ленточными маркерами (tape mark) и которые мы будем обозначать звездочкой (\*). Структуру файла на магнитной ленте условно можно изобразить так:

HDR1\* данные \*EOF1\*

Как вы видите, после метки конца файла также записывается ленточный маркер.

Мы упоминали о метке начала тома (VOL1); покажем полную структуру ленты с одним файлом:

VOL1 HDR1\* данные \*EOF1\*\*\*

и с двумя файлами:

VOL1 HDR1\* данные \*EOF1\*HDR1\* данные \*EOF1\*\*\*

Нетрудно заметить, что конец ленты определяется по двум дополнительным ленточным маркерам.

Отметим, что такой признак конца тома (EOF1 и три ленточных маркера) всегда записывается при закрытии файла, а следовательно, любой записанный на магнитную ленту файл является последним; это очень важно помнить. Файлы могут оказаться недоступными только потому, что последний файл был записан в середину ленты.

И еще одно необходимо отметить. При перемотке и поиске информации на магнитной ленте основным разделяющим признаком является ленточный маркер, а дополнительными — метки начала и конца файла. Поэтому мы не можем перемотать ленту назад на один блок (управляя ею из KMON), несмотря на то что данные записаны поблочно. Можно вернуться только на одну (две, три и т. д.) позиции, занимаемые файлами, а затем, перемещая ленту в прямом направлении, по метке начала файла найти нужный файл.

Каталог магнитной ленты составляется путем чтения информации из меток VOL1, HDR1, EOF1. Вместо начального номера блока в листинг выводится последовательный номер файла, называемый позицией. Размер файла определяется путем подсчета количества блоков данных, которое за-

тем проверяется по данным в метке конца файла. Таким образом, лента, не содержащая никаких файлов, после инициализации имеет следующий формат:

VOL1 HDR1\*\*EOF1\*\*\*

Так как программы ОС, реализующие отдельные функции, должны находиться на устройстве с произвольным доступом, то мы пока не будем говорить о загрузке с магнитной ленты, а сразу перейдем к описаниям меток.

Каждая метка занимает 80. байтов (символов в коде ASCII) в 256.-словном физическом блоке, что соответствует одной строке на экране дисплея или одной перфокарте. Читать информацию удобно с помощью программы DUMP, руководствуясь таблицей значений байтов метки.

Данный стандарт определен для любых операционных систем, и, естественно, в метках есть поля, которые RT-11 не использует (например, срок хранения файла). Использование кода ASCII для меток, а также то, что они ограничивают файл с обеих сторон, позволяет определить характеристики файла, даже если предыдущие файлы испортились и программа DIR не может собрать их в каталог. Однако здесь мы не можем вносить запись о файле в каталог ввиду отсутствия такового. Но мы можем скопировать файл, если знаем его номер (позицию) на ленте.

Подчеркнем два важных момента.

На одной ленте допускается присутствие нескольких файлов с одним именем и типом, что может привести к некоторым затруднениям при копировании, если используется имя файла. В этом случае в качестве указателя лучше использовать его позицию.

Эффективность использования логических дисков при записи на ленту можно достаточно точно рассчитать. Размер каталога — 8 блоков, а каждый файл на ленте требует дополнительно по 2 блока для меток (HDR1 и EOF1). Таким образом, если мы хотим записать на ленту более 5 файлов, то лучше организовать из них логический диск и записывать его одним файлом. Дополнительные расходы памяти на ленте для каталога окупаются отсутствием меток начала и конца файла. В дальнейшем мы увидим, что эффективность использования логических дисков при записи на магнитную ленту еще выше.

*Продолжение следует*

## РИФ

## Экранный редактор программ на основе алгоритмического языка для интерпретатора Фокала

62

Переход от составления алгоритмов на алгоритмическом языке пробного учебного пособия к написанию программ на языке программирования — один из сложных моментов при преподавании машинного варианта курса ОИВТ. Обойти его позволяет Е-практикум, не дающий, однако, возможности написать диалоговую программу, а тем более программу на конкретном производственном языке программирования. Организичный переход от учебных языков к учебно-производственному языку Рапира обеспечивает система «Школьника». Конструкции Рапиры близки к соответствующим конструкциям алгоритмического языка. Однако эта система реализована пока лишь на ЭВМ типа СМ и «Агат». Что же касается Бейсика и Фокала, то для них, насколько нам известно, подобных систем пока нет.

В [1] изложен метод программирования на производственных языках на основе алгоритмического языка. По сути дела, в этой работе рассмотрен метод структурной записи программ на языках, не имеющих достаточных средств структурирования, путем использования в качестве «структурного каркаса» языка, располагающего такими средствами. Особенно эффективным оказалось использование этого метода в школьной информатике, причем языком структурного каркаса здесь является алгоритмический язык. В [1] введено понятие базового языка, который является формализованным расширением алгоритмического языка (в основном за счет операторов ввода-вывода производственного языка, в частности Фокала). Программа обычным образом записывается на базовом языке в правой части листа, а затем построчно кодируется слева операторами производственного языка. Поскольку распространенные версии Бейсика, Фокала и других языков допускают запись комментариев в одной строке с исполняемым оператором (правее его), а также в отдельной строке, команды базового языка, отсутствующие в производственном, могут быть «закрыты» для интерпретатора, т. е. закодированы как комментарии. Рассмотренный в [1] метод позволяет все этапы программирования (проектирование алгоритма, алгоритмизацию, собственно программирование и отладку программ) выполнять в терминах базового языка, независимо

от того, каким интерпретатором оснащена данная ЭВМ.

Для этих целей разработана программа РИФ — экранный Редактор программ для Интерпретатора Фокала. Он позволяет вводить программы на алгоритмическом, точнее на базовом, языке в ЭВМ и кодировать их прямо на экране дисплея. РИФ содержит упрощенный набор команд экранного редактирования, позволяющий выработать у учащихся некоторые умения и навыки, предусмотренные как действующей, так и перспективной программой школьного курса информатики. Наконец, этот редактор является этапом в разработке автоматизированного учебного редактора-конвертора с базового языка на производственный. Подобная система позволит в будущем производить автоматическое кодирование или ручное кодирование под ее контролем (для обучения).

Основные команды редактора. РИФ разработан для микро-ЭВМ ДВК-1 с дисплеем «Электроника 15ИЭ-00-13». Команды, кодируемые клавишами ↑, ↓, →, ←, управляют перемещением курсора во всех направлениях, имеются также команды перемещения курсора в начало текущей и следующей строк, в левый верхний угол экрана. Команды ↔, ←→ вызывают соответственно вставку пробела и удаления символа в позиции курсора. Вставка новой строки (содержащей номер) ниже позиции курсора производится командой ↑, удаление строки — командой ↓. В редакторе имеются команды, позволяющие вставить в текст редактируемой программы заготовки различных конструкций алгоритмического языка. Например, при последовательном нажатии клавиш СИЕ в текст программы будет вставлена заготовка

```
5.05 С      ЕСЛИ      ;
5.10 С      ТО        ;
5.15 С      ;          ;
5.20 С      ИНАЧЕ    ;
5.25 С      ВСЕ       ;
```

Имеются эффективные средства для просмотра текста программы: команды перемещения в окне экрана на одну строку вверх и вниз, «перелистывания» вперед и назад, вывода на экран первой «страницы» текста, вывода последней строки. Текст самого редактора в обычном режиме для



просмотра и модификации недоступен. РИФ позволяет вести отладку программы, не удаляя его самого из ОЗУ. Он имеет команду запуска отредактированной программы на исполнение, команду выхода из редактора. Выделена специальная клавиша для повторного исполнения последней введенной команды.

**Ввод и редактирование программ.** Ввод новой программы начинается с того, что в редакторе с помощью команд типа  $\uparrow$  или C1 (буква) формируется текст, состоящий из заготовок команд базового языка и строк пробелов. Затем выполняется редактирование введенных строк: заготовки превращаются в полные команды, вводятся остальные необходимые команды, имена величин и т. д. При попытке вставки новой строки команда не будет выполнена, если соседние с позицией курсора строки имеют подряд идущие номера, не позволяющие вставить между ними новую занумерованную строку. В этом случае выводится сообщение «ИЗМЕНИТЕ НУМЕРАЦИЮ СТРОК». Подобная ситуация изображена на рис. 1. Курсор находится в строке 1.18, следующая строка имеет номер 1.19. Редактор отказывается выполнять между ними вставку. Изменить нумерацию можно например, исправив 1.19 на 1.20.

Переставить строку с одного места на другое можно комбинацией двух команд. Переставляемую строку следует удалить командой  $\nabla$ , а затем, переведя курсор в нужное место, дать команду вставки строки. При этом будет вставлена только что удаленная строка, но уже с новым номером. Дело в том, что при удалении строки попадают в стек программных строк

РИФа, а при вставке извлекаются из него. Таким образом, можно переставлять одну или несколько строк, манипулируя с верхними элементами стека: последовательно одну за другой удаляя эти строки, а затем в обратном порядке вставляя в нужные места. При последовательных удалениях (вставках) курсор автоматически занимает нужную позицию.

Программа, изображенная на рис. 1, введена полностью. РИФ выводит ее на экран дисплея точно в таком же виде, как с помощью оператора Фокала WRITE ALL. Однако с точки зрения интерпретатора Фокала она состоит лишь из одних комментариев: в каждой строке после номера стоит символ С (COMMENT). Завершается каждая из строк ограничителем строки — ;. Он проставляется РИФом и связан с особенностями реализации. Ни удалить его, ни ввести что-либо правее ограничителя строки невозможно.

Далее программу следует закодировать операторами Фокала. На рис. 2 представлен результат кодирования введенной программы. Выполняется редактирование строки 2.35. Осталось ввести номер 2.95 перехода по третьей альтернативе оператора IF. Содержащиеся в строке 2.35 служебные символы  $\blacksquare$  означают: первый — что перед вводом слова ПОКА была нажата клавиша перехода на русский алфавит РУС; второй соответствует клавише ЛАТ. Вывод таких символов на экран обеспечивает взаимнооднозначное соответствие между кодами сохраняемой в памяти ЭВМ строки текста и строкой, выводимой на экран дисплея. Это создает дополнительные удобства при редактировании данной строки. Аналогичным свой-

63

РЕДАКТОР ПРОГРАММ РИФ		
1.01 С	АЛГ ЧИСЛО СОЧЕТАНИЯ	;
1.04 С	НАЧ НАТ М, N, A1, A2, A3	;
1.11 С	ASK "M=", M, "N=", N	;
1.16 С	ФАКТОРИАЛ (N, A1)	;
1.17 С	ФАКТОРИАЛ (M, A2)	;
1.18 С	ФАКТОРИАЛ (N-M, A3)	;
1.19 С	TYPE "C=", A1/(A2*A3)	;
1.95 С	КОН	;
1.99 С		;
2.05 С	АЛГ ФАКТОРИАЛ (НАТ К, А)	;
2.06 С	АРГ К	;
2.07 С	РЕЗ А	;
2.08 С	НАЧ ЦЕЛ I	;
2.20 С	A=1	;
2.30 С	I=0	;
2.35 С	ПОКА I<K	;
2.40 С	НЦ	;
2.45 С	I=I+1	;
2.50 С	A=A*I	;
2.55 С	КЦ	;
2.95 С	КОН	;

♦♦ИЗМЕНИТЕ НУМЕРАЦИЮ СТРОК♦♦

РЕДАКТОР ПРОГРАММ РИФ

1.01 C	АЛГ ЧИСЛО СОЧЕТАНИЯ	;
1.04 C	НАЧ НАТ M,N,A1,A2,A3	;
1.11 ;	ASK "M=",M,"N=",N	;
1.16 S K=N;DO 2;S A1=A;C	ФАКТОРИАЛ (N,A1)	;
1.17 S K=N;DO 2;S A2=A;C	ФАКТОРИАЛ (N,A2)	;
1.18 S K=N-M;DO 2;S A3=A;C	ФАКТОРИАЛ (N-M,A3)	;
1.19 ;	TYPE"C=",A1/(A2*A3)	;
1.95 R	КОН	;
1.99 C		;
2.05 C	АЛГ ФАКТОРИАЛ (НАТ K,A)	;
2.06 C	АРГ K	;
2.07 C	РЕЗ A	;
2.08 C	НАЧ ЦЕЛ I	;
2.20 S	A=1	;
2.30 S	I=0	;
2.35 IF (I-K)2.40,2.95,2.95, _	В ПОКА I < K	;
2.40 C	ИЦ	;
2.45 S	I=I+1	;
2.50 S	A=A*I	;
2.55 GO 2.35;C	КЦ	;
2.95 R	КОН	;

ством обладает оператор строчного редактирования MODIFY интерпретатора Фокала, однако вывод служебных символов сразу во всех строках программы создал бы неудобства при ее чтении и отладке. РИФ обеспечивает вывод служебных символов лишь в текущей редактируемой строке.

Строка в окне экрана дисплея становится текущей редактируемой («открывается» для редактирования), если она имеет номер, в нее предварительно переведен курсор, нажата клавиша, соответствующая произвольному символу, который мы хотим ввести в данной строке. В любой момент может быть не более одной открытой строки, поэтому при появлении новой такой строки старая «закрывается». Закрытие строки происходит также при выполнении различных команд вставки и «перелистывании». При открытии строки раздается гудок и, если в строке есть русские буквы, строка зрительно раздвигается: в ней появляются служебные символы. При переходе от одной строки к другой в момент ввода в нее символа автоматически происходит закрытие — открытие: старая строка закрывается и зрительно сжимается за счет удаления служебных символов (если в ней есть русские буквы), новая строка открывается и при необходимости раздвигается.

Редактировать можно всю строку, в том числе ее номер, за исключением зоны ограничителя — последней ;. При изменении номера строки надо руководствоваться правилами Фокала для номеров строк, т. е. номер должен иметь вид M.N, где M — номер группы, M=1, 2, ..., 99, N — номер строки в группе, N=01, 02, ..., 99. Кроме того, новый номер строки обязательно должен находиться в интервале, образованном номерами соседних строк (здесь номера рассмат-

риваются как действительные числа). В случае нарушения этих требований при закрытии строки восстановится старый номер и в нижней строке будет выведено сообщение:

**НОМЕР СТРОКИ M.N ИЗМЕНЕН НЕВЕРНО. ВОССТАНОВЛЕН СТАРЫЙ НОМЕР**

По окончании редактирования программу можно запустить, не выходя из редактора. В случае правильного завершения программы управление снова будет передано редактору. Это обеспечивается кодированием команды КОН основного алгоритма оператором R (RETURN) (рис. 2, строка 1.95).

Аппаратные и программные условия реализации. Разработка РИФа проводилась на базе учебного кабинета вычислительной техники ДВК-1/ДВК-2М. Оборудование такого кабинета рассмотрено в [2]. Он состоит из двенадцати машин ДВК-1, используемых в качестве микро-ЭВМ учащегося, и ЭВМ учителя ДВК-2М. В кабинете оборудована простая локальная сеть, в которой ДВК-1 связаны линиями последовательной передачи информации с ДВК-2М. ДВК-2М функционирует под управлением ОС ДВК (аналогичной РАФОС). ДВК-1 оснащены ПЗУ с интерпретатором Фокала. Все функции управления и обмена программой сосредоточены на ЭВМ учителя, откуда можно подключиться к любому ДВК-1, загрузить или выгрузить файл с диска главной машины в ДВК-1, переключить ДВК-1 на дисплей ДВК-2М, получив тем самым возможность просмотреть, исправить и выполнить программу учащегося с пульта ДВК-2М. Для исполнения в ДВК-1 можно переслать с диска ДВК-2М произвольную программу, написанную, например, на ассемблере или Паскале и скомпилированную в абсолютном формате (тип файла LDA в смысле РАФОС).

а также программу на Фокале. Однако двусторонние пересылки файлов (с диска в ДВК-1 и обратно) возможны лишь для программ на Фокале.

При таком составе аппаратных и программных средств кабинета разработка экранного редактора программ могла проводиться лишь на Фокале. РИФ должен храниться на диске ДВК-2М, перед началом урока рассылаться по сети на все ДВК-1, сохраняться в ОЗУ ДВК-1 в процессе разработки (с его помощью) и отладки программ. Если на одном из ДВК-1 работа завершилась, т. е. программа отлажена, и ее нужно сохранить, РИФ следует удалить из ОЗУ командами Фокала. Это приведет к освобождению памяти; затем следует загрузить отлаженную программу на диск ДВК-2М.

**Итоги и перспективы.** Сейчас РИФ эксплуатируется на уроках информатики и при проведении 8-часового межшкольного практикума в кабинете вычислительной техники школы № 314 Куйбышевского района Москвы. Все занятия проводятся методом программирования на структурном каркасе алгоритмического языка. Первые результаты показывают, что РИФ существенно повышает эффективность занятий, позволяя учащимся вводить в ДВК-1 программы, написанные на базовом (расширенном алгоритмическом) языке без кодов. Дальнейшее редактирование, кодирование на Фокал, исправление ошибок в процессе отладки выполняются прямо на экране дисплея. Автоматизируя множество рутинных операций при работе с программой, РИФ позволяет учащимся значительно быстрее достичь результата — получить работающую программу, а это,

в свою очередь, повышает их интерес к предмету.

Программа РИФ сама написана на структурном каркасе; в качестве базового языка использован язык Э. Дейкстры [3]. В нынешнем варианте она использует для работы практически всю доступную пользователю память ДВК-1 — 48К байт, однако при «сжатии» (удалении комментариев и изготовлении таким образом «загрузочного модуля») она, по-видимому, могла бы быть использована и на других микро-ЭВМ с интерпретатором Фокала, например, на «Электронике БК-0010».

В качестве перспектив укажем на планируемое расширение набора команд редактора, обеспечение возможности редактирования старых программ и разработку на его основе автоматизированного редактора-конвертора с базового языка на Фокал. Представляется также полезной разработка подобного редактора на ассемблере с последующей «зашивкой» его в ПЗУ совместно с интерпретатором Фокала.

Справки можно получить по тел. 460-19-80, Москва.

#### Литература

1. Штильман З. М., Штильман Б. М. Метод программирования на Бейсике и Фокале на основе алгоритмического языка // Микропроцессорные средства и системы. 1986. № 3.
2. Преснухин Л. Н., Фролов Г. И., Куправа Т. А. и др. Учебный класс на основе диалоговых вычислительных комплексов // Микропроцессорные средства и системы. 1985. № 3.
3. Дейкстра Э. Дисциплина программирования / Пер. с англ. М.; Мир, 1978.

М. БУНЯЕВ, И. ДАВЫДОВ  
МГПИ им. В. И. Ленина

## Автоматизированная система подготовки обучающихся курсов «Радуга»

В 1986 г. группа преподавателей и сотрудников математического факультета МГПИ им. В. И. Ленина и ИПИ АН СССР приступила к разработке методики применения диалоговых автоматизированных обучающих систем (АОС) в учебном процессе высшей школы.

Определены принципиальные направления разработок:

создание методик преподавания, ориентированных на применение ВТ;

создание собственной, оригинальной АОС,

исследование ее функциональных, дидактических возможностей и их экспериментальная проверка в практике обучения;

использование различных АОС в учебном процессе как средства управления самостоятельной работой учащихся при изучении учебного материала.

Схематизированная в АОС «технология» преподавания должна иметь достаточно гибкую структуру, широкий набор дидактических возможностей, что может быть достигнуто не за счет создания сверхсложной

системы, а использованием набора средств генерации обучающих программ с различными функциональными возможностями для создания автоматизированных обучающих курсов (АОК). Широкое применение АОК в практике преподавания невозможно, если в распоряжение преподавателя не будет передан удобный «инструментарий» — набор средств создания АОК (набор генераторов, редакторов обучающих программ) и учебно-методических материалов по их применению в учебном процессе. Наиболее перспективным мы считаем создание АОС на базе дисплейного класса, оснащенного локальной сетью ПЭВМ.

Группой разработана оригинальная система, в которой нашли воплощение сложившиеся у разработчиков представления о разумной организации диалога с преподавателем — составителем АОК и обучаемым — пользователем АОК. Это АОС «Радуга», предназначенная для подготовки обучающихся курсов и ориентированная на КУВТ «Ямаха». Система может, с незначительными изменениями, использоваться для ЭВМ другого типа.

Классические (краудеровские) идеи организации автоматизированного обучения предполагают такую схему: учащемуся сообщается некоторая порция информации, заканчивающаяся вопросом и несколькими возможными вариантами ответа. Он выбирает тот вариант, который считает правильным, после чего в зависимости от точности выбора ему предлагаются либо следующая порция информации, либо разъяснение характера ошибки, либо дополнительная тренировочная серия вопросов, либо материал для повторения (если характер ошибки свидетельствует о наличии пробелов в знаниях).

В процессе работы на АОК составитель устанавливает связь между вопросами в пределах каждого из фрагментов курса путем указания номера вопроса-наследника для каждого из предложенных вариантов ответа. Так как возможные варианты переходов от вопроса к вопросу определяются выбором варианта ответа на каждом шаге, то в результате работы возникает разветвленная, многоуровневая, частично адаптивная обучающая программа.

Заметим, что при создании АОК можно обойтись без предварительной проработки его логической структуры в целом. Достаточно продумать связь вопросов в пределах каждого из фрагментов и связь фрагментов между собой. Эта деятельность преподавателя аналогична обычной работе по составлению планов занятий, проводимых традиционными методами.

Каждый из фрагментов АОК является Бейсик-программой, что полностью снимает сложности при передаче его по локальной сети ПЭВМ и упрощает управление учебным процессом. Объем каждой программы не должен превышать 23К байта, чтобы не было нужды в использовании дисководов. Практика работы с системой показала, что допустимы программы, содержащие 10—15 вопросов с 5—6 ответами на каждый и подробными текстами реакций. Это более чем достаточно для организации работы с ЭВМ в рамках одного занятия. (Согласно рекомендациям предварительных физиологических и психологических исследований самочувствия обучаемых при работе на цветных дисплеях, длительность работы с каждым из фрагментов АОК ограничена 40—45 мин.)

Основным достоинством системы является то, что процесс создания Бейсик-программ, реализующих отдельные фрагменты АОК, не требует от преподавателя навыков в программировании и в обращении с операционной системой.

Работа каждой программы, созданной с помощью системы, состоит в последовательном выполнении следующих шагов:

вывод заголовка;

вывод очередного вопроса, получение ответа, вывод реакции на ответ и переход к следующему вопросу-наследнику;

вывод концовки.

Заголовок и концовка (первый и последний кадры вопросника) — это произвольные тексты, например: «Занятие 15. Тема: «Решение простейших тригонометрических уравнений», «Конец занятий» и т. п.

Вопрос — это произвольный текст, требующий выбора одного из предложенных вариантов ответа.

Вывод вопроса может быть выполнен с задержкой или без задержки ответа (это определяется составителем программы). Вывод с задержкой состоит в том, что после появления текста вопроса обучаемый вводит свой ответ и только после этого выводятся варианты ответа, предложенные преподавателем. При выводе без задержки варианты ответа появляются сразу же.

После появления на экране вариантов ответа обучаемый должен выбрать один из них. Это осуществляется установкой стрелки против выбранного варианта и нажатием на управляющую клавишу. На экране сохраняется текст вопроса, выбранный вариант ответа и появляется текст соответствующей ответу реакции. Как правило, это сообщение о том, что ответ верен или неверен (с разъяснением характера ошибки), или указание к решению, или полный ответ на

поставленный вопрос и т. п. После нажатия на управляющую клавишу, в зависимости от того, какой вариант ответа был выбран, происходит переход к следующему вопросу и т. д.

Если же на экране находится сообщение, не требующее ответа, то нажатие на управляющую клавишу приведет к переходу к следующему вопросу или сообщению.

Таким образом, при работе с программой обучаемый использует только три клавиши: управляющую («возврат каретки») и клавиши перемещения стрелки («курсор вверх» и «курсор вниз»); нажатие любой другой клавиши не оказывает никакого действия.

Основной редактор АОК написан на Паскале. После запуска редактора на экране появляется основное меню — список всех операций, которые он может выполнять. В правой части экрана находится стрелка. Чтобы выполнить одну из операций, нужно с помощью клавиш перемещения курсора установить стрелку в нужную строку и нажать управляющую клавишу.

Основное меню содержит названия девяти возможных операций: «Редактировать заголовок», «Редактировать вопрос», «Редактировать концовку», «Удалить вопрос», «Очистить память», «Проверить полноту», «Читать вопросник», «Писать вопросник», «Закончить работу».

Деятельность составителя фрагмента АОК сводится к последовательности следующих действий в предоставляемых для этой цели окнах:

- редактирование текста заголовка;
- редактирование текста очередного вопроса или сообщения;
- редактирование текстов вариантов ответа;
- редактирование текстов реакций на каждый из вариантов ответа;
- редактирование текста концовки.

Предоставляется возможность изменения цвета окон, их размеров, создания новых или уничтожения ранее созданных. Богатство цветовой палитры, предоставленное в распоряжение создателя обучающих курсов, и послужило причиной названия системы — «Радуга».

При работе системы в любой момент можно получить краткую подсказку о том, какое действие вызовет нажатие той или иной клавиши. Эта подсказка или находится в нижних строках экрана, или вызывается нажатием специальной клавиши.

Система автоматически проверяет «полноту» создаваемого вопросника, т. е. выдает сообщения о представленных вопросах и о наличии переходов к несуществующим вопросам.

В процессе работы система собирает ин-

формацию о вопросах, на которые отвечал обучаемый; о том, какой из вариантов ответа на каждый из вопросов был выбран; о числе возвратов к одному и тому же вопросу и т. п. Эта информация может быть использована преподавателем при оценке успешности обучения.

Работа с редактором в режиме «Меню» полностью избавляет преподавателя не только от необходимости составлять Бейсик-программу, но и от общения с той или иной операционной системой. Однако это ни в коей мере не исключает возможность усовершенствования уже созданных программ путем включения в них специально написанных программных блоков. Если возникла необходимость использовать в программе какие-либо непредусмотренные основным режимом возможности, например включить в качестве вопроса, ответа или реакции изображение, звук, видеофильм и т. п., необходимо включить в программу соответствующий фрагмент на Бейсике. Если при этом руководствоваться приведенными в инструкции к системе правилами, то сохранятся возможность создавать всю стандартную часть с помощью основного редактора.

С сентября 1987 г. кафедра математического анализа МГПИ приступила к проведению педагогического эксперимента по применению АОС «Радуга» в учебном процессе на базе трех групп первого курса в рамках обычного расписания.

Обучение ведется комбинированным методом: традиционное обучение сочетается с автоматизированным. Число часов, отводимых на проведение занятий традиционными методами, сокращено (лекции — на 1/4, практические занятия — на 1/3). Эти часы выделены для проведения экспериментальных занятий с применением АОС, которые включают: самостоятельную работу над теоретическим материалом, самостоятельную работу в рамках практических занятий, проведение контроля. Выделено необходимое время для выполнения домашних заданий и на внеаудиторную самостоятельную работу, управляемую системой.

Средства эксперимента — учебный класс, оборудованный 11 ПЭВМ «Ямаха», набор обучающих, контролирующих и тренировочных программ, подготовленных системой «Радуга» по всем разделам программы курса «Математический анализ» для пединститутов.

В рамках эксперимента удалось резко повысить объем самостоятельной работы студентов по изучению учебного материала с сохранением высокого качества обучения, усовершенствовать проведение текущего, ру-

бежного и итогового контроля знаний.

На этапе экспертных оценок обучающих программ, составленных в целях проведения эксперимента, рядом специалистов высказывалось опасение, что использование ответов множественного выбора, а не конструируемых ответов снизит педагогический эффект автоматизированного обучения. Однако, на наш взгляд, эти опасения необоснованны.

Во-первых, при современном уровне развития ПЭВМ и их математического обеспечения практически невозможно заставить машину «узнать», какой ответ был дан, если обучаемый написал или «сконструировал» свой ответ. По этой причине при создании АОС «Радуга» (как мы надеемся, всего лишь первой из серии систем этого класса, за которой последуют и более совершенные) было отдано предпочтение ответам по схеме множественного выбора.

68

Во-вторых, в некоторых случаях разница между конструируемыми ответами и ответами множественного выбора исчезает. Речь идет о тех случаях, когда круг разумных ответов строго ограничен небольшим числом возможных вариантов. Следовательно, возможность достижения высокого педагогического эффекта при использовании системы определяется четким выделением области ее применения в учебной деятельности. Заметим, что эта область оказалась достаточно широка.

В-третьих, оказывается, что в рамках одной и той же схемы функционирования (вопрос — ответ — реакция) можно формализовать достаточно много приемов обучения. С помощью таких схем в обу-

чающих программах дается теоретический материал с необходимыми разъяснениями, проверяется его усвояемость, проводится разбор примеров, приводятся упражнения и задачи для самостоятельного решения, разбираются домашние задания, проводятся контрольные работы, коллоквиумы, зачеты, составляются обучающие игры и т. д. Одним словом, набор этих схем покрывает многие формы деятельности преподавателя в процессе занятия.

Эффективность автоматизированных форм обучения будет во многом зависеть от творческой активности самих преподавателей в выявлении новых возможностей разрабатываемого им в помощь программного обеспечения. Наша же основная цель — дать каждому преподавателю достаточно удобный инструмент для реализации его собственных педагогических замыслов.

Несомненным достоинством системы «Радуга» является то, что простота структуры (отсутствие детерминированного, заранее обусловленного вида связей между вопросами) определяет ее высокую гибкость, возможность создания существенно отличных друг от друга по логической структуре обучающих программ. Система является удобным средством реализации методических замыслов преподавателя в процессе подготовки занятия.

В настоящее время эта система используется при подготовке обучающих курсов на факультетах МГПИ, в ряде педвузов СССР, в школах Москвы.

Адрес для справок: 107140, Москва, Краснопрудный пер., 14. МГПИ им. В. И. Ленина, математический факультет.

И. БРИК

## Простые программы на Фокале

Одна из проблем, стоящих перед учителем информатики, — дефицит программного обеспечения. Для педагога, работающего с КУВТ-86, положение усугубляется сравнительно малой распространенностью языка Фокал.

Однако некоторые несложные программы, например для контроля знаний учащихся, учитель способен составить сам. В этом ему помогут некоторые приемы программирования на Фокале, о которых говорится ниже.

### 1. Работа с символьной информацией

Серьезный недостаток языка Фокал — от-

сутствие возможности работы с символьными (литерными) переменными. И все же, используя функцию FCNR, можно смоделировать многие операторы и функции для работы с символьными переменными языка Бейсик.

```
10.10 T ! "ВАША ФАМИЛИЯ ?" ! " : " ; S K=0
10.20 S K=K+1 ; S P (K) =FCNR (FCNR (-1))
10.30 I (P (K) -13) 10.35, 10.50
10.35 I (P (K) -153) 10.20, 10.40, 10.20
10.40 F A=1, K ; S P (K) =0
10.45 T ! ! "ПОВТОРИТЕ ВВОД !" ; G 10.10
10.50
11.10 T ! , K-1
11.20 T ! ; F A=1, K-1 ; X FCNR (P (A))
11.30 T ! ; F A=5, 7 ; X FCNR (P (A))
```

Приведенный фрагмент программы позволяет в ответ на запрос ввести в машину с клавиатуры любую последовательность символов. При этом введенные символы отображаются на дисплее, и одновременно формируется соответствующий этой последовательности одномерный массив, состоящий из десятичных кодов вводимых символов. Они могут быть любыми — буквы, цифры, знаки, пробел. После окончания ввода последовательности нужно нажать клавишу «ВВОД» (ее код — 13).

При вводе последовательности символов нельзя пользоваться клавишами редактирования для исправления ошибки. Для исправления ошибки при вводе в программу включены команды 10.35—10.45. Если какой-то символ введен неверно, нужно нажать клавишу «СБР→» (ее код — 153) и ввести всю последовательность символов заново.

Введенный набор символов записан в память машины в виде массива кодов P(A), где  $A=1, \dots, K=1$ . Теперь мы можем работать с данным набором (символьной переменной) почти так же, как и на языке Бейсик. Например, распечатать этот набор (оператор PRINT) на Бейсике

11.10 T !; F A=1, K=1; X FCHR (P(A)) или определить количество символов в наборе функции (функция LEN на Бейсике)

11.20 T K=1

Можно распечатать часть набора, например три символа, начиная с пятого

11.30 T !; F A=5,7; X FCHR (P(A))

Можно также объединить два набора символов в один.

машина печатает на дисплее вопрос, варианты ответа и ждет ввода ответа в течение 10 секунд. При этом отсчет секунд ведется в правом верхнем углу дисплея и сопровождается звуковыми сигналами. Если в течение этого времени не будет нажата ни одна клавиша, машина печатает: «Вы просрочили время!» Если же ученик введет ответ до истечения этого срока, машина печатает оценку ответа («Правильно!» или «Неверно!»)

Как работает программа? В строке 10.10 задается код правильного ответа T-50 (50 — десятичный код цифры 2). Во время обдумывания ответа циклически с интервалом в одну секунду проверяется содержимое регистра клавиатуры. В этот регистр автоматически записывается код последней нажатой клавиши. Адрес регистра клавиатуры — 177662. Для работы программы необходимо, чтобы первоначально в этот регистр был записан десятичный код клавиши «ВВОД» (здесь ее код 10), т. е. последней клавишей, нажатой в диалоге перед выполнением фрагмента, должна быть «ВВОД». Дело в том, что регистр клавиатуры доступен только при чтении и недоступен при записи и программными средствами изменить его содержимое невозможно. Если до истечения 10 секунд содержимое регистра клавиатуры изменится, будет нажата клавиша ответа, ЭВМ выходит из цикла и анализирует ответ. Если же ответа нет, то ЭВМ выходит из цикла через 10 секунд и сигнализирует о просроченном времени.

Время на обдумывание можно задавать произвольно, меняя конечное значение параметра K в строке 10.10. Интервал в одну се-

```

10.10 X FCHR(12);T !!"ПЕРВЫЙ ВОПРОС";B T=50
10.20 T !!"НАЗОВИТЕ СТОЛИЦУ ФРАНЦИИ."!
!"1.РИМ!"!2.ПАРИЖ!"!3.ЛОНДОН!"!4.ТОКИО"
10.30 T !!"ВВЕДИТЕ НОМЕР ПРАВИЛЬНОГО ОТВЕТА;"
11.10 F K=1,10;S P=FX(1,177662);D 12
11.15 I (K-21)11.20,11.30
11.20 X FK(20,17);T "ВЫ ПРОСРОЧИЛИ ВРЕМЯ !";
G 13.25
11.30 X FK(20,17);I (T-P)13.10,13.20,13.10
12.10 F B=1,370;
12.15 I (P-10)12.30,12.40
12.30 S K=20;R
12.40 X FCHR(7);X FK(56,2);X 2.0,K
13.10 T "НЕВЕРНО !";G 13.25
13.20 T "ПРАВИЛЬНО !"
13.25 T !!"!!"ДЛЯ ПРОДОЛЖЕНИЯ НАЖМИТЕ "ВВОД" ";
X FK(32,11);A H

```

## 2. Контроль времени

При составлении различных программ контроля знаний или тестов полезно ввести ограничение времени на выполнение задания. В приведенном ниже фрагменте программы

кунду для отсчета времени (пауза) организуется с помощью «пустого цикла в строке 12.30. Оператор ASK в строке 13.25 в данном случае служит не для ввода значения переменной, он обеспечивает продолжение работы по программе при нажатии клавиши «ВВОД».

## Решатель задач Solver

Для облегчения решения расчетных задач во всем мире началась разработка объектно-ориентированных средств с элементами искусственного интеллекта, предназначенных для пользователей-непрограммистов, благодаря чему можно говорить о новом направлении в программировании. Вместо обычного директивного программирования применяется так называемый концептуальный подход [1]. От пользователя не требуется указания хода решения задачи, так как система сама найдет его на базе введенных в нее знаний, описывающих данную задачу. К этому классу относятся, например, система «TK!Solver» и разработанная в Институте кибернетики АН ЭССР система «Mico PRIZ», которую весьма успешно используют для автоматизации инженерных расчетов.

Программная система Solver разработана с целью демонстрации новых методов программирования и использования ПЭВМ на каждом уроке. Она выросла из реализованного на машине «Apple II» подмножества языка UTOPIST [2] и является средством создания абстрактных моделей объектов реального мира, которые состоят из переменных и соотношений между ними. Система Solver позволяет описать множества моделей и понятий и потом при наличии исходных данных провести вычисления и анализ соответствующих этим моделям объектов реального мира. Весьма полезной является возможность создания с ее помощью проблемно-ориентированных пакетов для какой-то прикладной области, например для решения геометрических задач. В качестве вспомогательной программы Solver можно применять на всех школьных уроках, где приходится иметь дело с решением уравнений, многократными вычислениями и т. д. Таким образом, Solver содействует применению ЭВМ во всем учебном процессе.

Solver как программа моделирования является хорошим средством для развития абстрактного мышления у школьников. Конструирование моделей объектов и процессов окружающего мира, их изучение и использование содействует более глубокому пониманию изучаемого.

Основным входом системы является множество понятий, состоящих из алгебраических уравнений общего вида (выражение = выражение), которые можно рассматривать как классы каких-то объектов. Система самостоятельно проводит проверку всех

конструкций, сообщает об ошибках и предлагает пользователю исправить их.

Рассмотрим понятие квадрата.

```
<Квадрат
P=4*a
S=a^2
d^2=2*a^2
```

Каждое понятие может быть использовано для создания конкретных моделей объекта, которые используются для вычислений. Например, две разные модели квадратов с именами K1 и K2 можно создать так:

```
K1 Квадрат S:=1
K2 Квадрат a:=K1.d
```

Здесь K2 является квадратом, сторона которого равна диагонали квадрата K1.

Понятия могут быть занесены в архив, т. е. сохранены на магнитном диске. Таким образом можно создать пакеты для моделирования разных типов объектов.

Система Solver полезна при необходимости многократного решения некоторого множества уравнений с разными входными данными. Уравнения моделей могут быть описаны в том порядке, который пользователь считает для себя удобным; порядок их решения Solver найдет сама. Она является непроцедурной в том смысле, что понятия и модели описываются на декларативном языке с использованием знакомых математических терминов. Таким образом, пользователь работает только с описанием моделей, не определяя метод анализа или решения.

В качестве несложного примера работы с системой рассмотрим ход решения следующей задачи. Пусть у нас имеется шарик с известным диаметром (например, 10 см) и плотностью ( $1000 \text{ г/см}^3$ ). Из него вырезают возможно большой кубик и из кубика — в свою очередь шарик. Найти массу этого шарика.

Если у нас на дискете не имеется подходящих понятий, то можно их ввести с клавиатуры.

```
<Кубик
V=a^3
d^2=3*a^2
>
<Шарик
V=(4/3)*(pi*r^3)
d=2*r
>
<ФизическийОбъект
m=Ro*V
>
```



Теперь будем создавать конкретные модели шарика с диаметром 10 см:

Ш1 Шарик  $d:=10$   
кубика, диагональ которого равна диаметру большого шарика:

K1 Кубик  $d=Ш1.d$   
маленького шарика, диаметр которого равен стороне кубика K1:

Ш2 Шарик  $d=K1.a$   
Отметим также, что шарик Ш2 является объектом с конкретной плотностью  $1000 \text{ г/см}^3$ :

Ш2 Физический Объект  $Ro:=1000$   
Теперь мы готовы задать вопрос о массе маленького шарика:

?Ш2.м  
и система выводит ход решения.

Ход решения:

$K1.d=Ш1.d$

$K1.d \rightarrow 10$

$d^{\wedge}2=3^*a^{\wedge}2$

$K1.a \rightarrow 5.773503$

$Ш2.d=K1.a$

$Ш2.d \rightarrow 5.773503$

$d=2^*r$

$Ш2.r \rightarrow 2.886751$

$V=(4/3)^*(\#pi^*r^{\wedge}3)$

$Ш2.V \rightarrow 100.76663$

$m=Ro^*V$

$Ш2.m \rightarrow 100766.63$

$\Rightarrow Ш2.m=100766.63$

В системе Solver применяется представление уравнений и моделей в виде графа,

на котором основан анализ и синтез вычислительных алгоритмов для решения задачи. В состав системы входят блоки ввода-вывода, работы с файлами, формирования моделей и соотношений, планировки хода решения, решения моделей. Для решения отдельных уравнений используется аналитический или численный метод, для решения систем уравнений — комбинированные численные методы.

Система Solver реализована на языке Си на ПЭВМ IBM PC XT под операционной системой MS—DOS 3.0 и без изменений работает на персональной ЭВМ ЕС-1840. Solver может с ограничениями работать и на микро-ЭВМ с объемом памяти 64К байт; таким образом, ее можно использовать и на 8-разрядной ПЭВМ. Благодаря этому она может применяться как для решения текущих инженерных задач, так и в качестве учебной программы на самых разных школьных уроках и вспомогательной программы в вузах.

71

#### Литература

1. Тыгуз Э. Х. Концептуальное программирование. М.: Наука, 1983.
2. Томберг Л. Л. Решатель вычислительных задач для микроЭВМ // Диалог Человек — ЭВМ: Материалы III Всесоюзной конференции. Серпухов, Институт физики высоких энергий. 1984. С. 101.

В. БОНДАРОВСКАЯ, А. КОВАЛЕНКО, С. МИРОНЧЕНКО, Н. ПОВЯКЕЛЬ  
Киев

## Автоматизированное рабочее место учащегося: эргономика и гигиена

Введение курса ОИВТ потребовало создания в средних учебных заведениях специализированных кабинетов, оснащенных ПЭВМ. Организация кабинетов связана с решением ряда гигиенических и эргономических проблем. Важнейшей среди них является выработка требований к интерьеру кабинетов, организации рабочих мест учащихся и преподавателя, техническим и программным средствам обучения с помощью ЭВМ, факторам внешней среды, режиму обучения с применением ЭВМ и проведению медико-гигиенических мероприятий по профилак-

тике нарушений здоровья учащихся при неправильном использовании ЭВМ.

В настоящее время насущно необходимым является проведение исследований психологических характеристик учащихся в процессе обучения с использованием ЭВМ; их психофизиологического состояния, физиолого-антропометрических характеристик, с учетом которых разрабатываются автоматизированные рабочие места; параметров микроклимата в кабинетах ВТ, эргономических характеристик технических и программных средств.

Несмотря на актуальность проблемы ор-

ганизации кабинетов ВТ, ее изучению посвящено сравнительно немного исследований. Специалисты по гигиене детей и подростков, по дефектологии, по коммунальной гигиене обеспокоены положением дел в области применения компьютеров в обучении [1, 2, 3]. Однако они не рассматривают весь комплекс вопросов, связанных с организацией кабинетов ВТ, решение которых создает условия для обеспечения нормальной работоспособности учащихся и снижения их утомляемости. Например, основное внимание авторы работ [2, 3] уделяют параметрам изображения на экране, рассматривая их вне связи, например, с условиями освещенности и характеристиками рабочего места.

Как показывает анализ отечественных и зарубежных исследований по проблеме «работа с видеотерминалами», все указанные задачи необходимо решать как комплексную проблему [4, 5, 6]. Более того, поскольку речь идет о системе образования, то основным системообразующим фактором должно выступать психолого-педагогическое обеспечение использования ЭВМ в учебном процессе.

В данной работе приводятся результаты психологических, гигиенических и эргономических исследований, связанных с организацией автоматизированных рабочих мест учащихся в кабинетах ВТ, которые проводились сектором эргономических исследований Киевского НПО «Горсистемотехника» и отделом медицинских информационных систем Института кибернетики им. В. М. Глушкова АН УССР на базе средних учебных заведений Киева и Москвы.

Для комплексного изучения кабинетов ВТ Институтом кибернетики АН УССР совместно со специалистами Киевского института общей и коммунальной гигиены им. А. Н. Марзеева МЗ УССР была разработана методика, предусматривающая обследование площади, интерьера и ориентации кабинета, особенностей вентиляции, размещения и освещенности рабочих мест учащихся, характеристик технических средств и их размещения на рабочем месте, эргономических и гигиенических параметров рабочей мебели, режима работы кабинета.

Обследование проводилось в кабинетах информатики 11 средних школ Киева.

Для регистрации и обработки данных разработан документ «Карта санитарно-гигиенического обследования кабинетов ВТ», положенный в основу базы данных, реализованной на ПЭВМ «Ямаха».

С помощью специально разработанной в НПО «Горсистемотехника» анкеты, которая включала вопросы об опыте и режиме

работы учащихся с компьютерами, а также об их субъективной оценке своего самочувствия, был проведен опрос 75 учащихся IX—X классов школы № 132 Киева (ПЭВМ «Ямаха», дисплеи комплекса ЕС 1040), 86 учащихся УПК при ИПИ АН СССР Москвы (ПЭВМ «Ямаха»), 168 учащихся IX—X классов школы № 171 Киева (ДВК-1).

Полученные данные были подвергнуты анализу с привлечением материалов Всемирной организации здравоохранения по работе пользователей с видеотерминалами [7], нормативно-технической документации по эргономике [8, 9, 10, 11, 12], нормативных актов Минздрава СССР [13, 14, 15], результатов отечественных и зарубежных исследований по проблеме «работа с видеотерминалами» [4, 5, 16, 17, 18, 19].

Данные были систематизированы и сопоставлены с соответствующими нормами и рекомендациями (см. таблицу).

Как видно из таблицы, в кабинетах обследованных учебных заведений, за исключением школы № 134, площадь, приходящаяся на одно рабочее место, соответствует рекомендуемой площади — не менее 4,5 м<sup>2</sup>.

Иначе обстоит дело с таким параметром, как освещенность. Практически все авторитетные источники рекомендуют освещенность рабочего места с видеотерминалом, равную 300—500 лк [4, 5, 10, 18]. Исходя из такой нормы освещенности, определяют и требования к яркости символов на экране монитора и контрасту между символами и фоном [4, 5, 7]. Отклонения от указанных значений неизбежно приводят к искажению параметров изображения на экране.

Оказалось, что в обследованных учебных заведениях практически везде нормы освещенности рабочих мест учащихся нарушены как в сторону существенного уменьшения (7 школ), так и в сторону существенного увеличения (4 школы), что является совершенно недопустимым и пагубно воздействует на зрение.

Проектирование рабочих мест, оборудованных видеотерминалами или ПЭВМ, относится к числу важнейших проблем эргономического обеспечения средств вычислительной техники [20]. Здесь главную роль играет разработка специализированной рабочей мебели, определение таких параметров, как высота рабочей поверхности, размеры пространства для ног, высота рабочего сиденья, наличие и размеры подставки для ног, требования к расположению документов на рабочем столе, в том числе наличие и размеры держателя документов, требования к фактуре поверхности рабочего стола, регулируе-

Результаты обследования рабочих мест учащихся

Эргономические и гигиенические параметры рабочего места	Номер школы Киева											Норма
	79	134	77	51	84	УПК*	171	132	57	145	32	
Тип ЭВМ	Ямаха	БК-0010	Агат	ДВК-1	Агат	Ямаха	ДВК-1	Ямаха	Агат	Агат	Агат	
Площадь, приходящаяся на одно рабочее место, м <sup>2</sup>	8,1	4,1	6	5,1	5,2	6,1	6,1	13,2	5,4	6,1	6,3	4,5
Освещенность рабочего места, лк	80—100	700—2000	150—900	20—170	30—240	50—230	400—3000	30—100	40—360	100—180	160—1100	300—500
Размеры стола, см												
высота	70	73	75	74	2×67**	75	78	74	75	78	74	регулируемая 63,5—74
ширина	68	68	60	65	30	70	100	74	69	55	60	минимальная 80
длина	120	125	70	120	120	120	110	125	122	122	105	80—120
Высота рабочего сиденья, см	40	43	35	43	2×40	48	47	45	42	41	41	регулируемая 43—50
Ширина рабочего сиденья, см	38	40	45	31	2×37	40	39	—	40	36	43	минимальная 40
Угол наклона спинки по отношению к вертикали, градусы	0	15	0	15	0	20	0	—	15	20	15	регулируемый 0—20
Возможность регулировки рабочего сиденья	—	—	—	—	—	—	—	—	—	—	—	должно регулироваться
Возможность регулировки высоты рабочей поверхности	—	—	—	—	—	—	—	—	—	—	—	должна регулироваться
Расстояние от глаз до экрана, см	50	50	—	40	80	58	71	40	43	40	—	40—70
Соответствие расположения монитора углу зрения	—	—	+	—	+	—	—	—	+	+	+	должно соответствовать
Соответствие перемещения монитора нормам	частично	частично	—	—	—	частично	—	частично	—	—	—	свободное перемещение по рабочей поверхности
Соответствие пространства для ног нормам	—	—	—	—	частично	—	—	—	—	—	—	глубина на уровне стопы — 60 см, колена — 40 см, высота минимальная 60 см
Высота клавиатуры, см	6	—	4,8	—	4,8	6	5,9	6	4,8	4,8	4,8	3
Возможность отделения клавиатуры	—	+	+	+	+	+	+	+	+	+	+	должна отделяться
Наличие держателя для учебных материалов	—	—	—	—	—	—	—	—	—	—	—	должен быть

\* УПК Днепропетровского района.

\*\* При обследовании столы и рабочие сиденья оказались трех типов, приведен один.

мость рабочего места и его элементов [4, 5, 20].

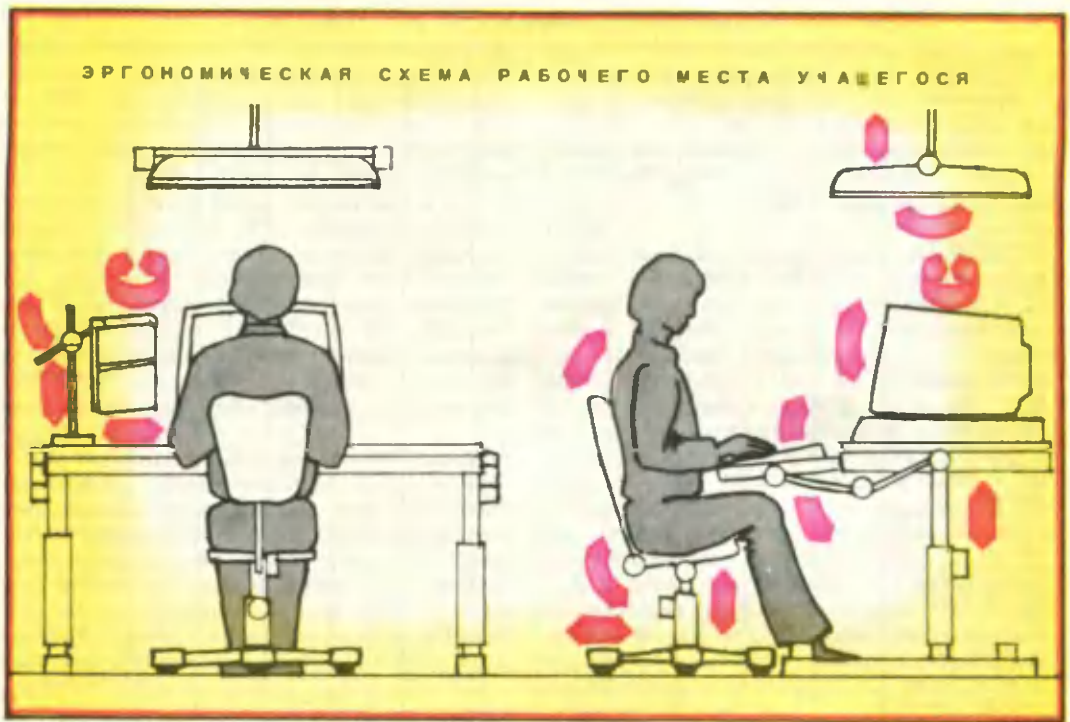
Как показали результаты обследования рабочих мест учащихся, ни одно из них не соответствует эргономическим требованиям. В кабинетах ВТ отсутствует специализированная мебель, рабочие места и их элементы не являются регулируемыми. Более того, в некоторых школах высота рабочей поверхности рабочего места учащегося выше предельно допустимого значения для взрослого пользователя (740 мм), что, естественно, приводит к искажению рабочей позы учащегося, а следовательно, и к мышечному утомлению, и к нарушению осанки. Не отвечают требованиям практически все параметры рабочего места. Рабочие столы в ряде

случаев узки, на них не помещаются технические средства; пространство для ног не соответствует требованиям, держатель для учебных материалов отсутствует, высота клавиатуры больше требуемой (30 мм на уровне среднего ряда кнопок), что приводит к еще большему нарушению требования к высоте рабочей поверхности.

На рисунке приведена эргономическая схема рабочего места учащегося, на которой стрелками указаны направления, в которых необходимо создавать возможности перемещения элементов рабочего места.

Сильным фактором зрительного утомления является наличие отражений от экрана монитора. Специалисты по эргономике видеотерминалов придают большое значение бор-

## ЭРГОНОМИЧЕСКАЯ СХЕМА РАБОЧЕГО МЕСТА УЧАЩЕГОСЯ



Эргономическая схема рабочего места учащегося  
1. Свободное пространство для расположения дополнительных периферийных устройств.

2. Свободное пространство для ведения записей.

Стрелками указаны направления перемещения элементов рабочего места.

бе с блескостью, бликами и отражениями [20]. Во многом наличие блескости и отражений зависит от фактуры поверхности экрана монитора, которая по общепризнанным нормам должна быть матовой, с коэффициентом отражения не выше 0,3. Для борьбы с бликами используют, как правило, специальные антибликовые покрытия.

Все обследованные нами технические средства обучения содержали экраны, не имеющие антиотражательных покрытий. Следовательно, зрительное утомление учащихся повышается.

Распространенным способом борьбы с блескостью и отражениями является рациональное расположение рабочих мест, оснащенных видеотерминалами или ПЭВМ, по отношению к источникам естественного и искусственного освещения.

Результаты нашего обследования показали, что во всех, без исключения, школах эргономические требования к расположению рабочих мест были нарушены. В меньшей степени это относится лишь к школе № 145, где при решении этих вопросов дирекция школы консультировалась со специалистами по эргономике. Как правило, мониторы на

рабочих местах учащихся расположены таким образом, что имеются прямые отражения от источников освещения, что категорически недопустимо.

В целом исследования показали, что вопросам эргономического и гигиенического обеспечения компьютеризации обучения уделяется явно недостаточно внимания.

Об этом свидетельствуют и результаты проведенного нами опроса учащихся. Режим использования компьютеров не регламентирован. Учащиеся работают с компьютером от 20—30 мин в неделю до 2—4 ч в день. И хотя они относятся к работе с компьютером положительно, тем не менее имеются жалобы на качество изображения на экране отечественных ПЭВМ, на неудобство рабочего места. Встречаются и астенопические жалобы — на мелькание перед глазами, головные боли, усталость глаз. Подобные жалобы наблюдались чаще всего у девочек, страдающих нарушениями зрения, особенно с нескорректированным зрением. А как отмечают офтальмологи, к работе за экраном нельзя допускать лиц с нескорректированным зрением [4, 5, 7]. Явления зрительного утомления могут быть следствием неправильного расположения рабочего места, появления отражений, а также неправильной освещенности рабочих мест.

Таким образом, в настоящее время одной из первоочередных задач компьютеризации

обучения является организация комплекса мероприятий по реализации эргономического и гигиенического обеспечения как рабочих мест учащихся, так и технических и программных средств обучения.

#### Литература

1. Гельтциева Е., Селихова Г. Режим работы за дисплеем // Информатика и образование. 1987. № 1. С. 82—92.
2. Труш В. Д. Компьютер и дети // Неделя. 1987. № 12 (1408). С. 9.
3. Маслов А., Таиров О., Труш В. Физиологические аспекты использования персональных ЭВМ в учебном процессе // Информатика и образование. 1987. № 4. С. 79—81.
4. Ergonomic Aspects of Visual Display Terminals / Ed. by Grandjean E., Vigliani E., L. 1980. 297 p.
5. International Scientific Conference: Work with Display Units. Stockholm, May 12—15, 1986. 1800 p.
6. Knave B., Wibom R., Voss M., Hedstrom L. а. от. Work with Visual Display Terminals among Office Employees // Scand J. Work Environment Health. 1985. V. 11N6. P. 457—466.
7. Visual Display Terminals and Workers' Health / World Health Organisation. Geneva. 1987. N 99.
8. ГОСТ 12.2.032—78. Рабочее место при выполнении работ сидя. Общие эргономические требования; Введ. 01.01.1979. М.: Изд-во стандартов, 1978. 9 с.
9. НМ МПК по ВТ 69—83. Машины вычислительные и системы обработки данных. Общие требования эргономики; Введ. 01.07.84. М., 1983. 23 с.
10. ОСТ 25 1205—85. ССЭТЭ СТ СМ ЭВМ. Дисплеи. Общие эргономические требования; Введ.

01.01.86. М., 1986. 11 с.

11. ОСТ 1206—85. ССЭТЭ СТ СМ ЭВМ. Органы управления. Средства индикации. Общие эргономические требования; Введ. 01.01.86. М., 1986. 10 с.
12. ОСТ 25 1217—85. ССЭТЭ СТ СМ ЭВМ. Рабочие места. Общие эргономические требования; Введ. 01.07.86. М., 1986. 10 с.
13. СНиП 11—65—75. Общеобразовательные школы и школы-интернаты. Нормы проектирования. М., 1975. 6 с.
14. ГОСТ 12.1.003—83. Шум. Общие требования безопасности; Введ. 01.07.84. М.: Изд-во стандартов, 1984. 9 с.
15. Гигиенические условия организации учебных занятий с применением компьютеров в средней общеобразовательной школе. Временные методические рекомендации/Минздрав СССР, ВНИИ гигиены детей и подростков Минздрава СССР. М.: Минздрав СССР, 1987. 15 с.
16. Бондаровская В. М. Человек и видеотерминал. Киев: Знание, 1985. 15 с.
17. Бондаровская В. М., Миронченко С. А., Полякель Н. И., Чачко С. А. Повышение потребительских свойств технических средств СМ ЭВМ в процессе их разработки // Приборы и системы управления. 1985. № 4. С. 18—20.
18. Димитрова Т. Инженерно-эргономично изследование на системата «човек — электронно-изчислителна техника» в учебния процес: Обзор. София, 1985. 66 с.
19. Phillips R., Burkhardt H., Coupland J. а. от. Computer Aided Teaching // Ergonomics. 1984. v. 27. N 3. p. 243—258.
20. Бондаровская В. М., Субботин А. Ю., Чачко С. А. Эргономические аспекты проектирования и эксплуатации видеотерминалов // Изменения, контроль, автоматизация. 1986. № 2 (58).

#### В. ДОСКИН,

докт. мед. наук, ВНИИ гигиены детей и подростков Минздрава СССР

#### С. КОЗЛОВСКИЙ,

канд. псих. наук, Институт общей и педагогической психологии АПН СССР

## Психометрическая оценка качества изображения на дисплеях ЭВМ

В настоящее время проблема оценки качества изображения, формируемого на экране дисплея, приобретает важное гигиеническое значение. Выполненная работа ставила своей целью показать практические возможности психометрических методов для решения этой проблемы.

Как показывают изучение литературы и наш опыт, у значительной части людей, которые по роду своей деятельности проводят у дисплеев большую часть рабочего времени, отмечаются неспецифические симптомы перенапряжения и утомления. Лица, работающие с компьютерами, жалуются

на слабость и головные боли, боли в области спины и шеи, зрительную усталость, двоение в глазах, раздражение и покраснение слизистой оболочки глаз, нарушение цветового зрения и др. Многие из упомянутых расстройств обусловлены низким качеством изображения на дисплеях ЭВМ. При длительном и систематическом восприятии изображений низкого качества жалобы на те или иные расстройства зрения, по данным разных авторов, отмечаются у 62—94 % работающих за дисплеем. Отрицательное влияние низкого качества изображения на организм человека происходит при совокуп-

ном воздействии целого ряда параметров изображения [1].

При высоком качестве изображения на дисплее число жалоб на те или иные расстройства зрения, снижение работоспособности и т. п. резко уменьшается.

Необходимость оценки качества изображения на дисплеях возникает при закупке ЭВМ в других странах, приемке новых отечественных образцов ЭВМ, при наладке и выбраковке работающей вычислительной техники. Однако в настоящее время количественная оценка качества изображения на дисплеях ЭВМ является трудной и практически недоступной большинству специалистов задач.

Для определения основных характеристик изображения (классов качества, оптимальных и допустимых значений отклонений от оптимального качества и т. п.) необходимо измерение его качества. При этом измерения могут быть проведены как инструментальными, так и психометрическими методами. Эти группы методов дополняют друг друга, выполняя различные функции.

Инструментальные методы позволяют измерить только единичные физические параметры качества изображения. Например, физический контраст изображения измеряется без учета влияния резкости на величину контраста, отношения сигнал / помеха и т. п. Следовательно, инструментальными методами нельзя измерить такой важный комплексный показатель, как ясность-сочность изображения [2], который связывает отдельные характеристики дисплеев и окружения рабочего места со специфическими характеристиками зрительной системы.

Наиболее точно сложные показатели качества изображения, учитывающие влияние нескольких единичных параметров качества, могут быть измерены психометрическими методами, при этом адекватным для измерения качества изображения является метод балльной последовательной оценки [2, 3, 4]. Измерения проводятся в два этапа: методом парных сравнений [5, 6, 7], а затем по балльной системе [4, 5]. Применение указанного метода дало хорошие результаты в технике кинематографии и телевидения, которые были использованы при разработке отраслевых стандартов [8].

В данной работе мы решали следующие задачи.

1. Оценить качество изображения, формируемого на экранах дисплеев различных типов ПЭВМ.

2. Сгруппировать дисплеи по классам качества изображений.

3. Установить (с учетом принятых в технике научной кинематографии и телевиде-

ния критериев) типы дисплеев, качество которых выходит за пределы допустимых значений.

Психометрические измерения были проведены для пяти типов дисплеев: дисплей ПЭВМ «Партнер» (Югославия); дисплей ПЭВМ «ДВК-2М», дисплей «ВТА-2000» (первый), дисплей «ВТА-2000» (второй), дисплей ПЭВМ «Агат» (СССР); дисплей «Мера» (Польша)\*.

Наличие двух одинаковых дисплеев послужило своеобразным контролем точности психометрических оценок.

На экраны дисплеев были выведены изображения букв русского алфавита (три горизонтальных ряда).

Экспертам было предложено оценить комплексный психологический показатель яркостных характеристик дисплеев. Этот показатель некоторые эксперты предпочитают называть ясностью-сочностью изображения. В работах [2, 9], посвященных качеству изображения, показано, что ясность-сочность изображения определяется такими физическими характеристиками как яркость, контраст, резкость, разрешающая способность, соотношение сигнал / помеха и др. Ясность-сочность изображения, являясь интегральным показателем качества, не учитывает, однако, имеющихся геометрических искажений. Кроме того, ясность-сочность изображения не имеет аналога на физической шкале измерений. Выбор этого показателя диктовался тем, что именно он, как совокупность яркостных характеристик изображения, оказывает влияние на функциональное состояние субъекта.

В качестве экспертов были приглашены 18 сотрудников ВНИИ гигиены детей и подростков, большинство которых по роду своей деятельности работали за различными ПЭВМ.

Указанные шесть дисплеев были расположены перед экспертами по вогнутой кривой. Экспертам были розданы бланки с инструкцией экспериментатора и первичной матрицей для записи результатов анализа качества изображений на дисплеях. В первичной матрице эксперт должен был отметить, какое из двух сравниваемых изображений ему больше нравится (все сравнения в этой серии исследований проводились попарно).

Далее на основе первичных матриц строилась матрица предпочтения (см. табл 1). Значение каждого элемента матрицы предпочтения соответствует доле экспертов, которые предпочитают изображение, номер

\* Дисплеи «ВТА-2000» и «Мера» являются терминальными устройствами ЭВМ СМ-4.

Таблица 1

Матрица предпочтения

	Дисплей ПЭВМ «Партнер»	Дисплей ПЭВМ «ДВК-2М»	Дисплей «ВТА-2000» первый	Дисплей ПЭВМ «Агат»	Дисплей «Мера»	Дисплей «ВТА-2000» второй
	1	2	3	4	5	6
1	—	0,33	0,33	0,20	0,27	0,43
2	0,67	—	0,56	0,20	0,33	0,97
3	0,67	0,44	—	0,13	0,27	0,56
4	0,80	0,80	0,87	—	0,43	0,73
5	0,73	0,67	0,73	0,57	—	0,80
6	0,57	0,03	0,44	0,27	0,20	—

которого стоит в столбце, изображению, номер которого стоит в строке.

Матрица предпочтения преобразовывалась в матрицу стандартного отклонения (чувственных различий или субъективных расстояний) на основе таблиц нормального распределения (см. табл. 2).

Поскольку значения элементов матрицы стандартного отклонения были определены на основе экспериментальных данных, то для установления шкальных оценок применялся суммирующий алгоритм [4, 6, 7]. Шкальная оценка равна среднему арифметическому по столбцу.

На основе парных сравнений удается достаточно тонко проранжировать уровни качества изображения. Вместе с тем этот метод не обеспечивает решения следующих важных для практики задач:

- определение отношения экспертов к уровням (состояниям) качества;
- обеспечение сравнительной оценки уровней;

определение значимости для экспертов субъективных расстояний между различными качествами изображений;

формирование на основе субъективных оценок классов качества изображений.

Для решения этих задач была проведена вторая серия психометрических измерений. Подробная методика этих исследований изложена в ряде опубликованных работ [2, 3, 4].

Экспертам в определенном порядке — от лучшего к худшему предъявлялись изображения, формируемые на дисплее, однако им это порядок был неизвестен. Лучшему изображению заранее устанавливалась оценка «5», а остальные изображения оценивались друг относительно друга: второе — относительно первого, третье — относительно второго и т. д. Эксперты могли поставить одну из следующих оценок: 5, 5 —, 4 +, 4, 4 —, 3 +, 3, 3 —, 2 +, 2. Знаки «+» и «—» означали при подсчете прибавление или, соответственно, вычитание 0,33 балла. Средние оценки качества изображения представлены в табл. 3.

Дальнейший анализ полученных экспериментальных данных позволил определить два класса качества, в которые входят рассмотренные дисплеи.

Исследователями установлено [2, 3], что каждым баллом эксперты оценивают не отдельную точку на шкале состояний (уровней) качества, а некоторый интервал этих состояний, названный нами интервалом ранжирования  $R_i$ . Интервал ранжирования  $R_i$  статистически оценивается в виде доверительного интервала, равного  $R_i = 2 \frac{t_{\beta} \sigma}{\sqrt{n}}$ , где

$t_{\beta}$  — параметр распределения Стьюдента,  $\sigma$  — средняя квадратическая ошибка,  $n$  — число экспертов.

Таблица 2

Матрица стандартного отклонения

	Дисплей ПЭВМ «Партнер» 1	Дисплей ПЭВМ «ДВК-2М» 2	Дисплей «ВТА-2000» первый 3	Дисплей ПЭВМ «Агат» 4	Дисплей «Мера» 5	Дисплей «ВТА-2000» второй 6
1	0,00	—0,46	—0,46	—0,85	—0,62	—0,17
2	0,46	0,00	0,42	—0,85	—0,44	0,94
3	0,46	—0,42	0,00	—1,13	—0,62	0,11
4	0,85	0,85	1,13	0,00	—0,17	0,62
5	0,62	0,44	0,62	0,17	0,00	0,85
6	0,17	—0,94	—0,11	—0,62	—0,85	0,00
Сумма	2,56	—0,53	1,60	—3,28	—2,70	2,35
Шкальная оценка	0,42	—0,09	0,27	—0,55	—0,45	0,41
Ранги	1	4	3	6	5	2

Таблица 3

	«Партнер»	«ВТА-2000» первый	«ВТА-2000» второй	«ДВК-2М»	«Агат»	«Мера»
Оценка, баллы	5	4,6	4,2	4,2	3,02	2,95

Согласно работе [2], вычисление  $R_i$  проводится для второго по качеству изображения (поскольку оценка лучшего изображения назначается экспериментатором и поэтому не имеет дисперсии) и в данном случае для первого из последующих изображений, не вошедших в первый интервал ранжирования. В рассматриваемом случае вычисление производилось для изображений, формируемых на экранах ПЭВМ «ВТА-2000» (первый) и «Агат».

78 Для уровня значимости  $P_{0,01}$ :  $t_{\beta} = 2,88$ , число экспертов  $n = 18$ ,  $\sigma_{\text{«ВТА-200» (первый)}} = 0,6$ ,  $\sigma_{\text{«Агат»}} = 0,62$ . Следовательно,  $R_{\text{«ВТА-200» (первый)}} = 0,8$ ;  $R_{\text{«Агат»}} = 0,81$ .

Так как  $R_i$  равно по 0,4 с обеих сторон от среднего значения, следовательно, в один класс качества входят получившие средние оценки от 5 до 4,2 балла: «Партнер», «ВТА-2000» (первый), «ВТА-2000» (второй), «ДВК-2М», а другой класс качества включает в себя ПЭВМ «Агат» и «Мера», их оценки составляют от 3,425 до 2,615 балла.

По нашим данным [2, 4] и по результатам зарубежных исследователей [10], изображения, оцениваемые ниже 3,75 балла, не могут быть использованы даже для показа художественных фильмов, а тем более в учебном процессе.

В заключение сформулируем выводы.

1. Разработанные ранее в технике кино и телевидения методы психометрических



измерений оказались продуктивными для оценки качества изображений, формируемых на дисплеях.

2. Психометрические методы позволяют проводить измерения как простых, так и сложных показателей качества изображения. Это особенно важно для оценки качества изображения на экране школьного дисплея, поскольку именно эти сложные показатели качества оказывают наиболее существенное влияние на зрительные функции растущего организма.

3. Психометрические измерения позволяют:

а) дать балльную оценку уровня (состояния) качества изображения на дисплее;

б) распределить дисплеи на классы в зависимости от качества изображения;

в) определить оптимальные значения и допустимые пределы отклонений качества изображения.

4. Психометрические измерения могут быть использованы для выборочного контроля качества дисплеев ПЭВМ, выпускаемых в нашей стране, а также для сравнительной оценки и отбора закупаемых в других странах.

#### Литература

1. Dainaff M. Visual Fatigue and Occupational Stress in VDT Operators // Human Factors. 1981. V. 23. N 4.
2. Козловский С. М., Истамина Е. И. Методы зонирования кинозалов // Техника кино и телевидения. 1973. № 3.
3. Козловский С. М. Методика ранжирования при оценке качества кинотелевизионных изображений // Труды НИКФИ. 1968. № 57.
4. Современные психометрические методы оценки качества кино и телевизионных изображений. М., 1978. 43 с.
5. Козловский С. М., Трофимов Б. В. Оценка качества изображения фильмокопий методом субъективной экспертизы // Техника кино и телевидения. 1972. № 11.
6. Guilford J. Psychometric methods. N. V., 1954.
7. Thurstone L. L. Fechner's Law and the Method of Equal Appearing Intervals // Journal of Experimental Psychology. 1929. N 12.
8. Строительные нормы и правила.— СНиП — П — Л — 68.
9. Stulz K. F., Zweig H. J. Roles of Sharpness and Graininess in Photographic Quality Definition // 1962. V. 52, N 1.
10. Prosser R. D., Allhatt I. W. Quality Grading of Impaired Television Pictures // Proc. JEE. 1964. V. 3. N 3.



**В. ЗБАРОВСКИЙ,**

канд. пед. наук, зав. кафедрой методики профтехобучения ВИПК РР и С ПТО  
**Б. МАРАТ,**

канд. мед. наук, доцент ЛГПИ им. А. И. Герцена

## Комплексное исследование использования ЭВТ

Современный уровень научных исследований требует комплексного подхода к решению проблем построения рационального режима учебно-воспитательного процесса.

В ВИПК профтехобразования в 1987 г. было проведено комплексное медико-педагогическое исследование эффективности применения ЭВТ в учебном процессе повышения квалификации директоров, инспекторов и методистов при изучении темы «Контроль за качеством и результатами обучения» по предмету «Управление профессионально-технической подготовкой».

Тема содержит ряд конкретных понятий, которые необходимо усвоить слушателям, а также статистические данные, требующие осуществления математических расчетов по заданной программе.

Основой создания сценария для ППС послужила проблемная лекция по этой теме.

С целью формирования творческого подхода к анализу результатов обучения в ППС предусматривается выбор оптимального варианта, поиск новых путей и способов определения уровня сформированности знаний, умений и навыков учащихся.

Важное значение для успешного применения ППС имеет поэтапный контроль и самоконтроль за усвоением учебного материала. С этой целью в программе предусмотрено в случае низкой оценки результата по этапу дополнительное разъяснение основных понятий и рекомендации по изучению литературных источников. Продвижение по программе от одного раздела к другому обусловлено уровнем усвоения материала; перейти к следующему этапу можно, только получив оценку не ниже «удовлетворительно».

Целью комплексного исследования является определение эффективности ППС и выявление изменений функционального состояния обучаемых в процессе работы с ЭВТ.

Сущность рабочей гипотезы заключалась в следующем:

обучение с использованием ЭВТ позволит сократить время на изучение темы и даст возможность слушателям лучше усвоить учебный материал при условии их напряженной работы;

длительная непрерывная работа с ЭВТ вызывает изменения функционального состояния обучаемых, что снижает эффективность обучения;

наибольшую нагрузку при обучении с использованием ЭВТ будет испытывать зрительный анализатор.

Результаты обучения с использованием ЭВТ сопоставлялись с данными, полученными при традиционном изучении содержания темы в аналогичных группах предыдущих потоков. Традиционное обучение предусматривало создание на занятиях проблемных ситуаций, дискуссии, беседы, т. е. активную деятельность слушателей по усвоению учебного материала.

Одним из показателей, характеризующих интенсивность обучения, является время, затраченное на изучение материала.

Анализ лекционных занятий по данной теме показывает, что активно работают в течение 90 мин не более 35—40 % слушателей, добросовестно записывают изучаемый материал 85—90 %, а 10—15 % слушателей не делают необходимых записей и не принимают участия в обсуждении изучаемых вопросов. В конечном счете, они не усваивают основное содержание темы.

Как показали наши исследования, обучение с использованием ЭВТ позволяет сократить время на изучение темы. По сравнению с традиционной методикой обучения продолжительность усвоения материала сокращается на 7—10 мин, в зависимости от уровня компьютерной грамотности обучаемых. При высоком уровне подготовки отдельных слушателей продолжительность изучения темы составляет 65—70 мин.

Другим показателем, характеризующим эффективность обучения, является индивидуальная работа на уроке. Индивидуальный темп работы позволяет лучше усвоить учебный материал, изучить основные понятия, выявить взаимосвязи и закономерности, осмыслить предъявляемую информацию. Результаты сравнительной контрольной проверки уровня усвоения учебного материала показали, что обучение с помощью ЭВТ существенно влияет на качество обучения. Слушатели лучше запоминают основные по-

нения, четко воспроизводят большинство из них, умеют применять знания в новых условиях.

Результаты сравнения ответов обучаемых представлены в табл. 1.

Таблица 1

Ответы слушателей	Число ответов, %	
	Традиционная методика (112 человек)	Обучение на ЭВТ (107 человек)
Полные правильные	32	52
Неполные правильные	46	42
Неточные	17	5
Ошибочные	5	1
<b>Итого</b>	<b>100</b>	<b>100</b>

80

При обучении с помощью ЭВТ большая часть учащихся дает правильные полные или правильные неполные ответы (94%), в то время как при использовании традиционной методики, включающей элементы активизации деятельности учащихся, такие ответы составляют 78%.

Рассмотрим результаты физиологических исследований при обучении с использованием ЭВТ.

Как видно из табл. 2, динамика показателей функционального состояния организма обучаемых в процессе занятий не была однозначной.

Выявлено статистически достоверное снижение критической частоты слияния световых мельканий, т. е. уменьшение функциональной подвижности зрительного анализатора, указывающее на нарастание зрительного утомления к концу занятия. Ухудшалось и общее самочувствие обследуемых, о чем свидетельствуют статистически достоверные различия интегральной оценки состояния до и после занятия. В процессе занятия практически не изменялся объем кратковременной памяти и наблюдалась тенденция к активизации функций внимания (как по объему перерабатываемой информации, так и по уменьшению количества допускаемых ошибок).

Приведенная динамика психофизиологических показателей может свидетельствовать о том, что использование ЭВТ в учебном процессе является фактором, повышающим мотивацию к обучению, мобилизующим психофизиологические резервы учащихся.

Однако нарастание зрительного утомления и особенно данные о существенном ухуд-

шении самочувствия обследуемых к концу занятия указывают на необходимость коррекции методики проведения учебного процесса при использовании ЭВТ с физиологико-гигиенических и с педагогических позиций.

Кроме того длительное обучение на ЭВТ может вызвать неблагоприятные явления, которые снижают эффективность последующих занятий, проводимых в течение учебного дня.

Как показали наши исследования, проведение в дисплейном классе двух занятий подряд по разным курсам приводит к отрицательному результату. Следует отметить, что в системе повышения квалификации высокое чувство ответственности не позволяет слушателям отказаться от участия в занятиях. Однако при этом возрастает число жалоб на утомление, особенно зрительного анализатора, просьбу заменить «рабочую» программу «игровой», сделать перерыв и т. д. Наблюдения показывают, что большинство обучаемых выполняют программу формально, сосредотачиваясь только в тех случаях, когда дальнейшее продвижение по программе обусловлено необходимостью обоснованного ответа или принятия решения на основе анализа предъявленных данных. Вместе с тем такая обязательность раздражает уставших обучаемых, что приводит в отдельных случаях к отказам работать с ЭВТ после первого часа второго занятия.

Длительная, напряженная работа в дисплейном классе оказывает отрицательное влияние и на последующие занятия, проводимые традиционными методами. Так, после четырех часов работы с компьютером слушатели группы инспекторов и методистов на следующем занятии были необычайно пас-

Таблица 2

Показатель	До занятий	После занятий
Внимание (информ. ед.): производительность (числитель)	18,4±2,0	21,6±1,9
	10,7±2,5	7,3±2,3
Ошибка (знаменатель)		
Объем кратковременной памяти (информ. ед.)	4,1±0,4	4,3±0,4
Критическая частота слияния световых мельканий (Гц)	36,7±1,3	33,0±1,7*
Самооценка состояния (баллы)	12,5±2,1	7,5±2,6*

Примечание: Звездочкой отмечены достоверные различия при уровне значимости 95%.

сивны, не могли четко ответить на поставленные вопросы, основные положения лекции просили повторить несколько раз. С целью активизации слушателей в ходе занятия преподавателю приходилось делать дополнительные перерывы, вводить элементы эмоциональной разрядки, ставить более простые вопросы, требующие краткого ответа на основе имеющегося у слушателей опыта.

Следует отметить, что аналогичное явление наблюдается и при обучении с помощью традиционных методов, при условии, что они создают повышенную нагрузку для обучаемых. В связи с этим необходимо всестороннее изучение данного феномена.

#### Выводы:

1. Поскольку для обучения с ЭВМ становятся все более характерными такие факторы, как повышение уровня сложности, на котором ведется преподавание, значительные потоки информации и ускорение изучения учебного материала, то с физиолого-гигиенической точки зрения существенное внимание должно быть уделено разработке рациональных вариантов режима деятельности учащихся в течение учебного дня и учебной недели, основанной на оценке психофизиологической стоимости их труда. Следовательно, вопросы физиолого-гигиенического нормирования времени непрерывной работы учащихся за экраном дисплея являются актуальными.

2. Одним из направлений оптимизации функционального состояния организма для профилактики переутомления учащихся должно стать совершенствование их двигательного режима в учебном процессе, в частности использование малых форм физической культуры как средства, снижающего напряжение механизмов регуляции физиологических функций, а также применение специальных упражнений.

3. Нам представляется важным целенаправленное формирование у обучаемых адекватных представлений о гигиенически целесообразном поведении в процессе учебной деятельности, в частности сообщение им необходимых эргономических сведений.

4. На основании исследования нами разработана структура урока, проводимого в дисплейном классе. Так как она используется в системе повышения квалификации взрослых, то при обучении учащихся профтехучилищ в нее необходимо вносить коррективы исходя из их возрастных особенностей.

**Организационная часть.** Распределение учащихся по рабочим местам, разъяснение требований к поведению в дисплейном классе, объяснение порядка обращения за помощью (1 мин).

**Вводная беседа.** Сообщение темы и цели занятия, разъяснение отдельных задач, показ приемов пользования клавиатурой, создание проблемных ситуаций по теме. В ходе беседы целесообразно обратить внимание на основные узловые моменты темы, предложить записать определения, данные, понятия в конспект, разъяснить причины типичных ошибок, дать четкие указания по выполнению задания (3 мин).

**Индивидуальная работа обучаемых с ЭВТ.** Самостоятельное выполнение заданий в соответствии с программой (20 мин).

**Обсуждение результатов законченного этапа (раздела).** Беседа по изученному материалу, постановка контрольных вопросов по проверке степени усвоения содержания раздела, выявление пробелов в знаниях, корректировка действий обучаемых (3 мин).

**Индивидуальное выполнение заданий на ЭВТ.** Обучаемые продолжают работать с ЭВТ в индивидуальном темпе (20+20 мин).

**Дополнительные задания.** Отдельные учащиеся заканчивают работу раньше, чем вся группа, поэтому необходимо направить их деятельность на помощь отстающим или выдать им дополнительное задание (10—20 мин).

**Подведение итогов.** Индивидуальное обсуждение результатов по мере выполнения задания. Общая заключительная беседа по пройденному материалу. Обсуждение степени достижения цели урока и решения поставленных задач. Задания для самостоятельной работы во внеучебное время (3 мин).

**Регламентированные перерывы.** Организация отдыха обучаемых посредством смены видов деятельности. Введение физкультпауз, функциональной музыки и т. д. (20 мин — 2 перерыва по 10 мин).

5. На основании исследования вычленены типичные трудности, с которыми сталкивается преподаватель, проводящий занятия с использованием ЭВМ:

необходимость поиска новых форм, методов и методических приемов, обеспечивающих усвоение учебного материала;

необходимость давать четкие инструкции по различным этапам темы (необходимость постоянного переключения);

недостаточная разработанность индивидуального текущего контроля за усвоением учебного материала;

недостаточное влияние на темп работы обучаемых;

сложность введения коррекций, устных пояснений;

ослабление контакта с аудиторией;

практическое отсутствие эмоционального воздействия.

## Бейсик проник еще в одно электронное устройство

В ЛГУ осуществлена реализация языка Бейсик на устройстве подготовки данных УПД ЕС-9051. Реализация предназначена для учебных целей: для обучения языку программирования и основам информатики на базе языка Бейсик, для написания обучающих программ, универсальных и предметно-ориентированных (вопросо-ответная система, экзаменатор, игровое обучение).

Система программирования на базе языка Бейсик позволяет также использовать УПД для экономических и инженерных расчетов. УПД ЕС-9051 представляет собой небольшую (с письменный стол) вычислительную установку с экраном 32 на 8 символов, двумя дисковыми и оперативной памятью 20К байт. В вычислительных центрах стандартно используется в качестве устройства автономной подготовки и ввода данных в комплекте с ЕС ЭВМ.

При недостатке вычислительной техники оснащенные транслятором с языка Бейсик устройства могут с успехом заменить микро-ЭВМ в учебном процессе.

В основе созданного в ЛГУ транслятора интерпретирующего типа лежит принцип концептуальной переносимости. Исходная версия транслятора написана на машиннонезависимом, ориентированном на типичные средства микро-ЭВМ языке дизайна ДИОР, близком к языку Бейсик. Кроме специально принятых мер для облегчения адаптации к типичному машинному языку, язык дизайна содержит большое количество инструкций, служащих целям надежности, документированности и мобильности.

Опыт конкретизации исходной версии в коды процессора УПД ЕС-9051 продемонстрировал перспективность разработанного подхода к созданию трансляторов.

Дополнительные сведения по реализации транслятора с языка Бейсик можно получить по адресу: 198904, Ленинград, Петродворец, Библиотечная пл., 2, НИИММ ЛГУ, И. П. Другову.

И. ДРУГОВ,  
В. ОРЖЕНОВСКИЙ

82

## Интерпретатор алгоритмического языка

В средней школе № 1 г. Харькова разработана программа «Интерпретатор алгоритмического языка». Интерпретатор написан на языке Бейсик Микро-80 и занимает объем памяти около 12К байт.

Принцип работы программы заключается в анализе каждого вводимого в режимах «Программирование» и «Редактирование» строкового выражения и формировании его Бейсик-эквивалента, который вносится в зарезервированную область памяти. Для введения и выполнения команд на алгоритмическом языке используется определенный набор директив. Интерпретатор реализует команды: присваивания, ветвления, повторения, выбора, повторения с параметром; позволяет выполнять арифметические и логические действия, производить вычисление

ряда функций, организовать работу со всеми типами величин.

Практически полная тождественность алгоритмического языка и его машинной интерпретации позволяет существенно повысить эффективность преподавания, так как учащиеся имеют возможность выполнить написанные на алгоритмическом языке программы непосредственно на ЭВМ. «Интерпретатор алгоритмического языка» может быть использован для ЭВМ, имеющих интерпретатор Бейсика.

Адрес для справок: 310002, г. Харьков, ул. Красина, 12, школа № 1, кабинет информатики. Тел.: 43-09-52, 43-12-86.

И. КРАСОВСКИЙ,  
В. ЛЫСЕНКО,  
Ю. ЦОФИН

В. ВЕЙЦМАН  
Таллин

## Старт в информатику

Вот уже несколько лет существует в школах предмет «Основы информатики и вычислительной техники». Но до сих пор не ясно: какова цель предмета, чему учить школьников?

К сожалению, предложенный нам учебник не дает ответов на эти вопросы. Выскажу мнение учителей-практиков: существующее учебное пособие — не помощник ни учителю, ни ученикам в изучении ОИВТ. Поэтому родилась идея создать иной, более эффективный вариант курса в рамках тех широких возможностей, которые предоставляет действующая программа. Такой курс создан и предлагается вниманию читателей. Скажу сразу, что школы, на которых этот курс «обкатывался», самые обычные, без «уклонов», не имеющие зачастую не только ЭВМ, но даже кабинетов информатики.

Предлагаемый курс ОИВТ для средней школы основывается на описанных ниже идеях.

Статус курса, его положение относительно других школьных предметов. По нашему мнению, ОИВТ — обычный общеобразовательный предмет. Во-первых, остальные предметы: математика, история, иностранный язык и т. п. ничуть не менее информатики важны для современного человека. Во-вторых, как прикажете связать «особую важность» предмета ОИВТ с мизерным количеством выделенных на его изучение часов и убогой технической базой?

Этот предмет, на наш взгляд, нельзя отнести ни к техническим, ни к гуманитарным дисциплинам. Он находится на стыке этих двух направлений и так или иначе связан со всеми науками.

Цель. Цель курса — заложить основы «информационной культуры» человека. Это означает: дать учащимся общее представле-

ние об информатике, современной электронно-вычислительной технике и способах ее использования. Таким образом, уровень знаний и умений учащихся, которого мы добиваемся, — это уровень грамотного пользователя ЭВМ, а не профессионала, работающего в области вычислительной техники. То скромное количество часов и те сверхскромные технические средства, которыми обеспечен новый предмет сегодня, не позволяют полностью решить эту задачу. Давайте будем реалистами. Сегодня (подчеркиваю — сегодня) есть смысл говорить лишь об общем представлении.

**Техническая ориентация.** Курс подразумевает использование ЭВМ и реализован таким образом, чтобы любая вычислительная техника, к которой возможен доступ, органично вписалась в его структуру.

**Структура.** В основе содержания курса — взаимодействие человека с программным обеспечением ЭВМ, основанное на представлении о функциональных связях узлов ЭВМ и представлении о логике компьютера.

Изложение материала идет «снизу вверх». Первоначально формируется представление о логике ЭВМ. Учащиеся знакомятся с основами алгоритмического подхода к решению задач, им даются некоторые основные понятия и термины информатики, сведения о функциональных связях узлов ЭВМ. Итоговая часть курса — ознакомление учащихся с программным обеспечением — зависит от возможностей доступа школьников к ЭВМ. Программа-максимум включает в себя работу с готовыми пакетами и создание собственных прикладных программ (например, обучающих или программ типа «электронный экзаминатор»).

В заключение курса даются сведения об-

зорного характера о наиболее распространенных сегодня компьютерах как в СССР, так и за рубежом, о тенденциях в развитии ЭВМ на ближайшие годы. Предлагается информация об основных направлениях использования современных ЭВМ (АСУ, САПР, АРМ, системах искусственного интеллекта), а также сведения по истории развития электронно-вычислительной техники. Программа-минимум охватывает только эту, заключительную, часть.

Материал курса состоит из основной части и приложений. Основная часть практически машинно-независима. Будучи принципиально против безмашинного варианта обучения, мы тем не менее учитываем сегодняшнюю реальность.

Приложения предлагаются для адаптации курса к конкретному типу ЭВМ, доступному данной школе. В приложениях изложены языки программирования Бейсик и Фортран-77, а также необходимый минимум средств создания программ на ЭВМ «Ямаха», на ЕС ЭВМ и на СМ ЭВМ.

Курс подготовлен в виде учебника. Материал разбит на две части, главы и параграфы. I часть (главы 1, 2) — материал для изучения в IX классе. II часть (главы 3, 4, 5) — материал для изучения в X классе.

## Часть I

Предлагаем вниманию читателей материал IX класса. Для каждой темы указано необходимое количество часов согласно разработанному поурочному плану. На вводном занятии, которое называется «На пороге мира компьютеров» (1 ч), происходит первое знакомство школьников с предметом. Раскрываются понятия «информатика», «компьютер», обозначается содержание предмета ОИВТ. Кроме того, в краткой, упрощенной форме излагается назначение ЭВМ.

### Глава 1. Введение в информатику (15 ч)

- |   |     |
|---|-----|
| 1.1. Алгоритм — основа логики машины                  | 1 ч |
| 1.2. Составление алгоритмов решения задач. Блок-схемы | 3 ч |
| 1.3. Типы данных                                      | 1 ч |
| 1.4. Операции над данными                             | 3 ч |
| 1.5. Данные в памяти ЭВМ                              | 3 ч |
| 1.6. Организация данных для ЭВМ                       | 1 ч |
| 1.7. Чтение файлов данных                             | 3 ч |

Основные усилия мы сосредотачиваем на развитии у учащихся умения поставить задачу перед исполнителем. Кто является исполнителем? В конечном счете — ЭВМ. Информатика и вычислительная техника настолько тесно связаны между собой, что «прятать» ЭВМ нам кажется неестественным.

Но к ЭВМ — исполнителю алгоритмов — ученика надо подводить постепенно, поэтому на первом этапе, при изучении тем 1.1 и 1.2, исполнителем алгоритма может быть кто угодно. Главное, на наш взгляд, заключается в четкой постановке задачи. Должна быть цель — решить задачу и средство достижения этой цели — алгоритм решения.

Содержанию задач уделяется особое внимание. Они придуманы и подобраны так, чтобы охватить самые различные темы — начиная от вычислительных задач и кончая приключениями героев А. Дюма. Опыт работы показал, что гуманитаризация задач (и всего курса в целом) порождает буквально взрыв интереса учащихся к предмету. На базе таких задач открываются широкие возможности для оживления урока диспутами, викторинами, соревнованиями и т. д. А это очень помогает усвоению материала.

На втором этапе, начиная с темы 1.3, учащиеся постепенно ориентируются на ЭВМ — исполнителя алгоритмов. В самой упрощенной форме они знакомятся с такими основными и широко используемыми понятиями информатики, как «данные», «типы данных», «массив данных», «структура», «файл». Происходит знакомство с операциями над данными. Речь идет об операциях, доступных пользователю ЭВМ: чтении, записи, сравнении и вычислениях.

Наконец учащиеся получают общее представление о механизмах обработки данных. С нашей точки зрения, это необходимо для понимания логики машинной обработки данных.

Задачи, предлагаемые учащимся на этом уровне, уже более строгие, поскольку исполнитель алгоритма определен однозначно — ЭВМ. Однако упор по-прежнему делается на задачи гуманитарного, прикладного содержания. Что касается вычислительных задач из курсов математики и физики, которыми пересыщен нынешний учебник, то они предлагаются только при рассмотрении действий ЭВМ над числовыми величинами. Их доля в общем количестве задач не превышает 20—30 %.

Для нас очень важную роль играет способ, которым реализуются алгоритмы решения задач. При изучении первой главы используются только блок-схемы. Почему? На наш взгляд, это наиболее наглядный способ записи алгоритма. А наглядность — необходимое условие, особенно на начальном этапе знакомства с компьютерной логикой.

В то же время псевдокод — алгоритмический язык, занимающий так много места в нынешнем учебнике, в качестве общего способа записи алгоритмов в принципе не нагляден. Это очень серьезный недостаток,

побудивший нас полностью отказаться от подобного способа записи алгоритмов на начальном этапе.

В самом деле, геологи, разведывающие незнакомую местность, имеют карту этой местности, но отнюдь не ее словесное описание. Точно так же ученики, вступающие в знакомый им ранее мир компьютерной логики, нуждаются в наглядной иллюстрации этой логики. В блок-схеме перед глазами картина всего алгоритма, налицо вся логика решения — что может быть лучше?

При изучении первой главы нет необходимости в регулярном использовании ЭВМ. На наш взгляд, в настоящее время это преждевременно и дает мало пользы, учитывая отсутствие соответствующего программного обеспечения. Зато отрицательные последствия такого раннего знакомства приходилось наблюдать в собственной практике. Ученики должны постепенно «созреть» для работы с компьютером. На это и направлен материал первой главы.

## Глава 2. Алгоритмический язык (17 ч)

- 2.1. Грамматика алгоритмического языка 1 ч
- 2.2. Быстрое начало (введение в структуру алгоритмического языка) 2 ч
- 2.3. Описание данных в алгоритмическом языке 1 ч
- 2.4. Операции над данными в алгоритмическом языке 2 ч
- 2.5. Организация данных в памяти ЭВМ 2 ч
- 2.6. «Вход» и «выход» компьютера. Обработка файлов данных 3 ч
- 2.7. Циклы 3 ч
- 2.8. Организация программы на алгоритмическом языке. Подпрограммы 3 ч
- 2.9. Дополнительные средства алгоритмического языка (для факультативного изучения)

Прежде всего — об идее алгоритмического языка. Она принципиально отличается от той, которая заложена в алгоритмический язык нынешнего учебника.

В первой главе мы вводим учащихся в логику ЭВМ. Настраиваясь на эту логику, на «алгоритмическое мышление», учащиеся использовали блок-схемы. Как уже говорилось, такой способ записи алгоритма решения задачи — общий и в принципе не зависит от исполнителя алгоритма.

Теперь введение окончено. Следующий этап — описание задачи для ее решения на ЭВМ. Вот тут необходим алгоритмический язык как способ записи алгоритма для ЭВМ. Таким образом, в нашем курсе нет принципиального различия между понятиями «алгоритмический язык» и «язык программирования». С единственной, правда весьма су-

щественной, оговоркой. Поскольку речь идет об учебном курсе информатики, то учебный алгоритмический язык, по нашим представлениям, должен удовлетворять следующим требованиям:

а) он должен быть ориентирован на реализацию как алгоритмов обработки данных (в первую очередь), так и алгоритмов решения вычислительных задач, т. е. язык должен быть универсальным;

б) он должен иметь достаточно простой синтаксис, близкий к естественному;

в) он должен содержать все основные логические конструкции, присущие современным, широко распространенным языкам программирования, оставаясь при этом «открытым» для новых конструкций;

г) правила алгоритмического языка не должны зависеть от особенностей конкретной ЭВМ;

д) он должен быть реализован на современных ЭВМ, чтобы ученики могли на практике проверить результаты своей работы.

Иными словами, это как бы учебная модель реальных языков программирования, иллюстрация, необходимая для общего представления о них. Если в дальнейшем возникает необходимость использования реальных языков программирования, такой переход будет быстрым и легким.

Алгоритмический язык, изложенный в нынешнем учебнике, не удовлетворяет двум важнейшим требованиям из вышеперечисленных. Во-первых, он не позволяет записать алгоритмы обработки данных, поскольку не имеет средств ввода-вывода и полностью игнорирует понятие файла. Значит, информация для алгоритмов организована неизвестно как, поступает неизвестно откуда, а результаты решения попадают в никуда. Во-вторых, этот язык не реализован и вряд ли может быть реализован в принципе, поскольку претендует на роль общего способа для описания любых алгоритмов. Видимо, этот недостаток ощущают и сами авторы языка, так как на протяжении вот уже нескольких лет все мы являемся свидетелями довольно неуклюжих и малоуспешных попыток все-таки реализовать его на ЭВМ и тем самым объять необъятное.

В силу этих причин мы отказались от использования алгоритмического языка учебника и пошли по иному пути. Было решено взять за основу один из популярных языков программирования и на его базе определить конструкции и правила учебного алгоритмического языка.

Вопрос о нотации языка, вызывающий столько споров, мы решили исходя из следующих соображений. Поскольку сама идея обработки информации подразумевает ее раз-

нообразии и в конечном итоге, интернационализацию этого процесса, вряд ли здесь можно считать обоснованными те жесткие ограничения, на которых настаивают авторы нынешнего учебника. Это порождает лишь проблемы с реализацией языка и путаницу в случае перехода к реальным языкам программирования. Английская нотация алгоритмического языка кажется нам более логичной, чем русская. Это косвенный способ повышения интереса к изучению иностранного языка. Ведь в данном случае учащиеся используют его на практике. И наконец, снимается вопрос о специальной адаптации языка в национальных школах союзных республик.

86

Итак, какой из существующих языков программирования выбрать за основу? Из наиболее распространенных — имеются в виду Бейсик, Паскаль, Кобол, Фортран и ПЛ/1 — только последний, на наш взгляд, имеет все необходимое, для того чтобы стать базой учебного алгоритмического языка. Его универсальность и поистине безграничные (пожалуй, даже чрезмерные) возможности позволили нам предложить язык КАЯ, удовлетворяющий всем перечисленным требованиям. КАЯ — аббревиатура от слов «компьютерный алгоритмический язык». (Кстати, это еще и распространенное в Эстонии женское имя.)

Следует отметить, что КАЯ — это не ПЛ/1! Искушенный программист не найдет в нем подводных камней и головоломок, за которые ПЛ/1 справедливо критикуют. «Чрезмерные возможности» ПЛ/1 позволили избежать также излишней сложности и машинно-ориентированных особенностей языка и, кроме того, сохранить его структуру открытой (т. е. в КАЯ можно добавлять новые конструкции «за счет» ПЛ/1). «Родство» заключается лишь в том, что алгоритмы для ЭВМ, записанные на языке КАЯ, без каких-либо ограничений или изменений обрабатываются любым транслятором языка ПЛ/1. Во всем остальном КАЯ — совершенно самостоятельное «детище», отлично справляющееся с теми задачами, которые вызвали его появление.

Скажем отдельно несколько слов о последней теме второй главы. Она предлагается для факультативного изучения и содержит расширение языка КАЯ в сторону инструментального средства для создания прикладных программ. Здесь изложены сложные логические конструкции проверки условий, в том числе конструкция «выбор», вывод данных в различных форматах. Причины, по которым включен этот материал, связаны с разработкой курса X класса.

Хотя изучение алгоритмического языка

преследует прежде всего теоретические цели, весьма полезно провести небольшое количество практических занятий. Вполне достаточно 6 часов за полугодие. Лучше всего, если практика будет внеклассной, скажем 2—3 экскурсии на ВЦ, во время которых учащиеся будут иметь возможность запустить заранее подготовленные программы и получить результат, а заодно и приобрести некоторые навыки общения с ЭВМ. Возможности для работы с языком КАЯ есть в принципе на всех типах ЭВМ, поскольку его базовый язык ПЛ/1 реализован на больших ЭВМ (в операционных системах ДОС ЕС, ОС ЕС и СВМ ЕС), на мини-ЭВМ (есть реализация для операционных систем ОС РВ и ДОС СМ), а также на микро-ЭВМ (реализация для популярной операционной системы СР/М). Однако мы рекомендуем ориентироваться на многопользовательские машины, прежде всего на ЕС ЭВМ.

ЕС ЭВМ, как правило, имеют терминальные классы с большим количеством дисплеев (15—20 и больше). Это позволяет организовать практику сразу для целого класса. В математическое обеспечение ЕС ЭВМ обязательно включается транслятор языка ПЛ/1, следовательно, не возникает проблем при работе с алгоритмами, записанными на языке КАЯ. Большая память и мощные процессоры ЕС ЭВМ позволяют учащимся в процессе работы получить представление о работе одной ЭВМ с несколькими пользователями практически одновременно. Ориентация же только на персональные компьютеры, как показала наша практика, ограничивает кругозор учащихся. И наконец, немалую роль играет широкая распространенность ЕС ЭВМ.

Предвидя упреки в ориентации на ЕС ЭВМ, еще раз хочу подчеркнуть, что речь идет только о факультативных практических занятиях для закрепления материала главы 2 и всего курса IX класса. С этой точки зрения, на данном этапе ЕС ЭВМ могут успешно использоваться. И уж, конечно, это лучший выход, чем сидеть сложа руки и ждать, пока в каждую школу придут персональные компьютеры.

Впрочем, те, кто совсем не имеет выхода ни на какие ЭВМ, могут успешно использовать язык КАЯ как теоретический материал, иллюстрирующий логику ЭВМ и способы компьютерной обработки информации.

В заключение несколько слов о педагогических программных средствах поддержки данного курса.

Третий год успешно используются две небольшие программы к материалу главы 2. Первая распечатывает в необходимом количестве карточки, содержащие тот набор клю-



чевых слов языка КАЯ (с транскрипцией и переводом на русский язык), который учитель собирается дать на очередном уроке. Вторая программа распечатывает в необходимом количестве карточки для контрольных работ по усвоению этих ключевых слов. Карточка содержит фамилию ученика, класс, небольшое шутивное приветствие и список ключевых слов в русском переводе. Против каждого слова — пустая графа, в которую ученик должен вписать соответствующий английский эквивалент.

Обе программы работают на ЕС ЭВМ под управлением операционной системы ОС ЕС, занимают минимальный объем оперативной памяти и написаны на языке КАЯ. Обе программы могут успешно использоваться в курсах иностранного языка и национальных языков.

В настоящее время проводится работа над пакетом обучающих программ к главе 1 пред-

лагаемого курса. Пакет будет работать в КУВТ «Ямаха» и, мы надеемся, в КУВТ «Корвет».

Автор статьи — системный программист — третий год ведет предмет ОИВТ в 51-й средней школе г. Таллина, является членом методического совета кабинета информатики ТГИУУ.

Предложенный курс подготовлен в виде учебного пособия и в настоящее время успешно используется в таллинских школах с русским языком обучения. В настоящее время готовится перевод курса на эстонский язык для использования в национальных школах Эстонской ССР.

Подробные справки можно получить по адресу: 200011, г. Таллин, ул. Вене, д. 5. Городской институт усовершенствования учителей. Кабинет ТСО и информатики. Гительман И. В.

Г. СКОБЕЛЕВ

## Наш эксперимент

Введение курса ОИВТ в школу сразу же поставило перед учителем вопрос: на какой технической базе обучать компьютерной грамотности учащихся отдаленных сельских школ, не располагающих ЭВМ и не имеющих близкой перспективы для создания кабинета, оборудованного компьютерами?

В качестве одного из выходов в свое время был предложен так называемый безмашинный вариант, предполагающий «меловое» изучение дисциплины и экскурсию в вычислительный центр. Однако, по нашему мнению, такой подход совершенно неэффективен: у школьника быстро исчезает интерес к теоретическому обучению, а практических навыков программирования такой ученик иметь не будет. Между тем существует довольно простое решение этого вопроса.

Кафедра методики преподавания математики Могилевского педагогического института по согласованию с Минпросом СССР и Минпросом БССР в 1986/87 учебном году организовала рассчитанный на 2 года эксперимент по выяснению целесообразности технической ориентации существующей программы по ОИВТ на программируемые микрокалькуляторы (ПМК) в тех школах Могилевской области, которые не располагают ПЭВМ.

В ходе эксперимента решались следующие задачи: выяснение того, насколько сознательно учащиеся усваивают основные принципы

программирования, пользуясь ПМК; смогут ли учащиеся приобрести прочные навыки программирования на ПМК за отводимое по программе время.

Эксперимент проводился в десятых классах девяти сельских и пяти городских школ области на ПМК «Электроника МК-61» (МК-54). Школам была предложена следующая программа:

Основные сведения о ПМК. Использование стековой памяти при вычислениях.	3 ч
Работа ПМК в режиме «Программирование». Блок-схема или алгоритмический язык как средство описания алгоритма. Ориентация алгоритма на машинное решение задачи. Программа.	3 ч
Программирование линейных вычислительных процессов. Проверка правильности программ.	2 ч
Программы с разветвлениями.	3 ч
Использование подпрограмм.	2 ч
Программирование циклических вычислительных процессов.	5 ч
Команды косвенных переходов и косвенного обращения к адресуемым регистрам.	3 ч
Решение задач.	13 ч

Перед началом эксперимента мы разработали и передали в школы методические мате-

риалы, в том числе пособие для учащихся, провели семинар для учителей. В пособии по каждому пункту программы было предложено достаточное количество задач как с решениями, так и для самостоятельной работы.

В течение первого года эксперимента сотрудники кафедры посетили уроки, консультировали учителей, приглашали их на семинары и проводили анкетирование.

За это время учащиеся научились реализовывать на ПМК основные вычислительные процессы: линейные, разветвляющиеся, циклические, не вызвало трудностей и использование подпрограмм.

При работе над линейными программами мы предложили ученикам баллистическую игру, основанную на применении известной из курса физики формулы вычисления расстояния, пролетаемого телом, брошенным под углом к горизонту, и пришли к выводу, что игры повышают интерес школьников к работе

с ПМК. Однако количество игр, по нашему мнению, должно быть ограничено, в противном случае учащиеся отвлекаются от изучения более сложных вопросов.

В эксперименте обращалось внимание на выяснение доступности для учеников использования косвенной адресации и возможности применения таких программ на уроках математики для построения графиков. Эти программы оказались доступными для учащихся и нашли практическое применение при построении сложных графиков.

Итоги первого года работы в эксперименте дали утвердительный ответ на вопрос о целесообразности применения ПМК на уроках ОИВТ в десятых классах. Использование ПМК позволяет формировать у учащихся навыки программирования и органически вписывается в школьный курс математики и физики.

# ЧТО? МОЖЕТ ЭВМ

## Параллельные компьютеры: стратегия внедрения

Многопроцессорные компьютеры с параллельной обработкой данных продолжают внедряться в оборонную промышленность и коммерческую сферу США. Однако этот процесс сталкивается с противодействием крупных корпораций, полагающих, что они получили еще недостаточно прибылей из компьютерных технологий, в которые их капиталы были вложены раньше. Недостаточно используются параллельные компьютеры и в научных иссле-

дованиях — тут препятствием становится их огромная вычислительная мощность и высокие цены, делающие их для многих организаций недоступной и изысканной роскошью.

Управление научно-технической политики при Белом доме разрабатывает план правительственных мер, направленных на расширение практического использования машин нового поколения. Главное в этом плане — создание национальной сети ЭВМ. Ее абонентами смогут стать те, кому сейчас суперкомпьютер не по карману; коллективность использования обеспечит достаточно высокую загрузку машин.

Предполагается, что линии передачи данных будут способны передавать 300 млн. бит в секунду — в 1000 раз больше, чем все существующие ныне локальные сети, вместе взятые. Возможные затраты — от 1,5 до 2,5 млрд. долларов на ближайшие 5 лет.

Второй путь требует значительно большего быстродействия ЭВМ, но и обещает больше. Потенциальные его возможности таковы, что огонь на поражение можно открывать сразу после первого пристрелочного выстрела, даже не дожидаясь разрыва пристрелочного снаряда. Для этого достаточно точно и быстро определять начальные условия полета снаряда и вводить их в компьютер.

## Влияние на экономику

В 1986 г. объем услуг, предоставляемых в США, превысил объем производства товаров более чем в два раза. Услуги составили 71 % валового национального продукта; их оказанием занято 75 % работающих американцев.

Довольно неожиданно: оказывается, Америка — страна сервиса, а не промышленности! Но еще неожиданной третья цифра:  $71\%/75\% = 0,947$ . Производительность труда в сфере услуг лишь на 5,3 % ниже, чем в экономике в среднем.

Это достигнуто благодаря широкому внедрению в обслуживающие фирмы ЭВМ, банков данных и информационных сетей. Каким именно образом компьютерные технологии позволили сервису встать вровень с промышленностью, в заметке не расскажешь — для этого нужна статья. А вот некоторые следствия применения ЭВМ отметить можно.

Во-первых, «кризисоустойчивость»: если обрабатывающие отрасли промышленности во время спадов увольняют работников, то отрасли сервиса продолжают, хоть и медленно, увеличивать занятость. Во-вторых, в американской экономике в последние годы быстро снижается доля рабочих мест с доходом ниже среднего.

Вот такое влияние ЭВМ!

Журнал в журнале  
для школьников, студентов,  
учащихся СПТУ и техникумов

Издается при участии ЦК ВЛКСМ

---

# МОЛОДЕЖНАЯ ИНИЦИАТИВА

---

# 4

**В НОМЕРЕ:**

---

● КУРСАНТЫ ВОЕННОГО УЧИЛИЩА ОСВАИВАЮТ ПРОГРАММИРОВАНИЕ

---

● НАШ КОРРЕСПОНДЕНТ СООБЩАЕТ

---

● РЕЦЕНЗИЯ НА КНИГУ ДЛЯ ЮРИСТОВ... И НЕ ТОЛЬКО ДЛЯ НИХ

---

● О НОВОЙ КНИГЕ ИЗДАТЕЛЬСТВА «МИР»

---

● ВНИМАНИЮ ПОЛЬЗОВАТЕЛЕЙ БК: ОТКРЫВАЕМ НОВУЮ РУБРИКУ

---

## Первая ласточка

Изучение компьютерной техники началось для нас около двух лет назад, когда мы — тогда еще курсанты первого курса факультета журналистики Львовского высшего военно-политического училища — слушали вводную лекцию по новому предмету «Основы устройства и применение электронно-вычислительной техники». Весь курс укладывался в 60 учебных часов. Но и за это сравнительно небольшое время преподаватель Н. Андреев сумел заинтересовать многих.

Персональные компьютеры привлекли нас в первую очередь как обязательный в будущем элемент журналистской работы. Появились первые публикации, набранные и отредактированные на дисплее, отпечатанные на принтере компьютера. К слову, этот материал полностью написан с помощью ДВК-2М. Вскоре зародилось желание не только пользоваться чужими программами, но и создавать свои.

Идея первой программы возникла так. Те знания, которые мы получали, хотелось применить на практике. Свою первую работу решили связать с одним из наших основных предметов. Техника производства и оформления газеты подходила для этой цели больше всего. Интересно было бы попробовать делать расчеты при макетировании газеты, решать задачи по полиграфии.

В первую очередь мы определили назначение программы и отобрали необходимые для включения в нее темы.

Эта первая серьезная работа, за которую мы взялись, шла параллельно с изучением Бейсика. Руководителю нашей работы офицер Ю. Коссовский помог нам в изучении специальных функций.

Но вот программа заработала, появился первый вопрос. К этому времени мы уже многое освоили в Бейсике, стали широко использовать различные подпро-

граммы. Через подпрограммы пришлось задавать такие операции, как очистка экрана, адресация маркера и некоторые другие, которые в более совершенных версиях Бейсика задаются операторами. Причем нередко мы мучались, создавали «невозможное», а потом обнаруживали это «невозможное» в одном из справочников. Но нет худа без добра. Хотя работа над нашей первой программой «Газета» затянулась, именно эти трудности способствовали более полному изучению компьютера и азов программирования.

Почему мы остановили свой выбор на Бейсике? Действительно, этот язык зачастую довольно резко критикуют более «подкованные» программисты. Но нас привлекло его существенное преимущество — возможность быстрого написания фрагментов программ и их отладки. Бейсик тем и хорош, что позволяет сразу получить более или менее работоспособный результат.

Итак, что же представляет собой программа «Газета»? Ее первоначальная цель — изучить три учебные темы, а именно кегль шрифтов в пунктах, расчет объема набора и расчет формата клише, после чего решить по ним задачи. В процессе работы над программой, посоветовавшись с преподавателями, мы решили дополнить ее одним наглядно-видовым вопросом: графический расчет клише.

Начинается программа с небольшой заставки, позволяющей «оживить» символьно-цифровой дисплей ДВК. Ее сменяет титульный лист, после которого по желанию можно вызвать на экран правила работы с программой или сразу перейти к работе. Важное качество — доброжелательность программы. Во время диалога компьютер обращается к



обучаемому по имени, тактично указывает на ошибки, поздравляет с успешным ответом.

Перед началом работы «Газета» выводит страницу со списком тем (в порядке возрастания сложности) и предлагает выбрать нужную. После появления на экране рамки с названием темы программа уточняет, приступать ли к решению задач, или вернуться к списку тем. Задачи построены на материале, охватывающем практические вопросы полиграфии, и задаются случайным образом. По мере успешного решения условия задач усложняются; например: в первых двух задачах обучаемый имеет дело с целыми числами, затем с дробными. Если в ответе допущена ошибка, программа сообщает об этом и предлагает ту же задачу повторно. После двух неверных ответов подряд на экран выводится справочный материал и программа ждет ввода от обучаемого сообщения о готовности к дальнейшей работе. По каждой теме нужно правильно решить пять задач, после этого появляется страница с оценкой за вопрос и сообщением о количестве правильных и ошибочных ответов.

Перед решением более сложных задач по расчету формата клише или объема набора предоставляется возможность ознакомиться со способом их решения.

Когда работа близилась к завершению, мы столкнулись еще с одной трудностью. Обучаемый, сидящий за клавиатурой электронно-вычислительной машины, был вынужден производить расчеты на листке бумаги. Пришлось внести в программу изменения, и теперь, если обучаемый запрашивает помощь компьютера, условие задачи перемещается в верхнюю часть экрана, а ниже на



выбор предлагаются формулы. Выбрав какую-либо из них, обучаемый вводит данные из условия задачи и получает правильный результат. Но для этого нужно знать, по какой формуле производится расчет в каждой задаче, т. е. понимать учебный материал.

В «Газете» предусмотрена возможность исправить оценку за учебную тему — для этого надо опять вызвать ее и решить все задачи. Но вот они решены, и обучаемый сообщает о прекращении работы. На экране появляется страница с общей оценкой за сеанс обучения, кстати, если она меньше «4», можно попробовать поработать снова. А отличнику приготовлен сюрприз — программа просит ввести более полные сведения о себе и выдает... отпечатанный на принтере «Диплом».

Пробная «обкатка» показала уязвимость «Газеты» — в большинстве случаев обучаемый, первый раз работающий с компьютером, за короткий срок успешно «забивает» программу, при этом нарушается нормальный ход ее работы. Со временем

изучение возможностей Бейсика позволило защитить «Газету» от ошибок ввода (случайно нажатая клавиша, символ вместо цифры, неверный аргумент для функции или математического действия). Подпрограмма защиты самостоятельно обрабатывает ошибку, исключая вывод на дисплей сообщения Бейсика об ошибке на английском языке, что, разумеется, снизило бы ценность самой программы. Внесен также и запрет прерывания работы программы нажатием (СУ/С).

Теперь «Газета» представляет собой защищенную программу, которую можно включить в состав автоматизированной системы обучения полиграфии. Для нас же она была первой ласточкой.

Мы использовали версию Бейсик/ОС ДВК 02—030, а программа «Газета» занимает 41 блок дисковой памяти.

**Н. ГАФУТУЛИН,  
И. ПАЩЕНКО**  
курсанты 2-го курса  
факультета  
журналистики ЛВВПУ

## Семинар в Киеве

Недавний визит М. С. Горбачева в СФРЮ показал, что у наших стран есть огромные возможности для контактов в самых разных областях. Одним из ведущих направлений, несомненно, является сотрудничество в области образования и компьютеризации.

23 апреля от Киевского вокзала в Москве отошел поезд, многие пассажиры которого вошли в вагоны со словами «компьютер», «школа», «комсомол». Это были участники предстоящего в Киеве семинара «Компьютеры и образование». Еще в пути они начали споры об актуальных проблемах, ставших затем предметом пяти специальных дискуссий между представителями ЦК ВЛКСМ и «Народной техники» Югославии (НТЮ).

«Народная техника» — массовая общественная югославская организация — объединяет людей, заинтересованных в повышении своей технической культуры. Каждый десятый югослав работает в одном из подразделений НТЮ. Практически в каждой школе чувствуется ее влияние.

«Народная техника» Югославии является составной частью Социалистического союза трудового народа Югославии и коллективным членом Союза социалистической молодежи Югославии.

В 1980 г. организации НТЮ перешли на принцип коллективного руководства с избранием председателя сроком на один год, членов Президиума на два года.

В последние годы одним из определяющих направлений деятельности НТЮ стали компьютеризация и информатика. «Народная техника» поддерживает тесные контакты с рядом югославских

фирм, выпускающих электронно-вычислительную технику.

...Столица Украины испытала гостей «на прочность» метеоусловиями. А всё, что зависело от хозяев, было организовано очень хорошо, и участникам семинара оставалось только четко изложить свои позиции в спорах, что охотно сделали практически все. Ведь темы дискуссий: «Молодежь и компьютеризация: вклад в ускорение научно-технического прогресса», «Компьютеры в системе средней школы», «Компьютеры в высшем и среднем специальном образовании», «Внешкольное компьютерное образование», «Вычислительная техника для образования» — волнуют одинаково и советских и югославских педагогов, представителей общественных организаций, журналистов.

Выяснилось, что многие проблемы (разнотипность техники, недостаток качественных программ) — общие для двух стран. Только в Югославии несколько раньше начали массовую компьютеризацию школы и поэтому продвинулись дальше нас. 70 % школ в этой стране оснащено вычислительной техникой (у нас — в среднем 30 %).

Семинар проходил с 25 по 28 апреля. В первый день пребывания на украинской земле участники возложили цветы к памятнику В. И. Ленину. На открытии семинара выступил секретарь ЦК ЛКСМ Украины В. Барабаш. В работе участвовало как с советской, так и с югославской стороны большое количество специалистов, заинтересованных в расширении творческих связей друг с другом.

Югославская делегация привезла с собой небольшую выставку персональных компьютеров, чтобы разговор был более предметным и предложения носили конкретный характер. Среди экспонатов были микро-ЭВМ «Орао», персональный компьютер «Лира».

Программа семинара была чрезвычайно насыщенной. Кроме дискуссий участников ждало посещение Института кибернетики им. В. М. Глушкова АН УССР, Киевского государственного университета Т. Г. Шевченко, Института электросварки имени Е. О. Патона.

Одно из самых неожиданных впечатлений. В киевской средней школе № 222 гостей ознакомил с постановкой преподавания информатики ученик IX класса Максим Ткаченко, способный программист. Так получилось, что в школе ожидали приезд ребята, а не взрослой делегации, — дети должны были по замыслу школы встречать гостей. Момент импровизационности, как говорится, имел место, но надо было видеть с каким достоинством, знанием дела отвечал Максим на вопросы гостей.

Максим рассказал, что кабинет вычислительной техники был создан в их школе три года назад, одним из первых в Киеве. Через полгода появился и кружок информатики. Девятые и десятые классы помимо обязательных посещают и факультативные занятия. Здесь кроме Бейсика изучают Паскаль. Старшеклассники составляют для младших школьников обучающие программы. Это говорит о высоком уровне подготовки в школе и, конечно, о нехватке программ в стране. Вместе с ребятами гости посидели за дис-

## Юристы — тоже люди

плями, на общем профессиональном языке (хотя и с помощью переводчика) поговорили о своих проблемах. Я заинтересовалась, знают ли в школе о существовании нашего журнала — в ответ мне представили ребят, которые его выписывают!

Интерес у югославской стороны в процессе дальнейшей работы семинара вызвал рассказ В. Фомина, директора компьютерного центра «Юность», готовящего к открытию на ВДНХ СССР. Ведь этот центр, организованный по инициативе ЦК ВЛКСМ, ГКВТИ, АН СССР и ВДНХ СССР скоро будет иметь оборот в один миллион рублей в год.

Так что возможно внешне-экономическое сотрудничество на компенсационной основе, обмен техникой, программным обеспечением. В перспективе — поездки на олимпиады по информатике в гости друг к другу, обмен публикациями в наших журналах («Народная техника» издает девять специальных журналов и газет). П. Радичевич, секретарь НТЮ, в беседе со мной отметил, что в Югославии не полностью загружены полиграфические мощности, есть бумага и в течение 1—2 месяцев для них не составило бы труда отпечатать любую нашу книгу или журнал сто тысячным тиражом. Интересная информация?

Сотрудничество только начинается. Климат доверия создан. Дело за специалистами.

Утверждение заголовка кажется вам тривиальным? Но из него можно сделать менее тривиальный вывод: что хорошо для юристов, то может оказаться подходящим и для всех остальных. В данном случае речь идет о книге «Микрокомпьютеры для юристов»\* и о самих микрокомпьютерах.

Судите сами: эта книга «...даст вам основы знаний о принципах работы современного компьютера, разъяснит смысл общепринятых терминов и концепций с тем, чтобы вы поняли ту реальную пользу, которую может принести компьютер. Вы найдете здесь описание компьютерных программ, разработанных специально для юристов: комплексный учет времени, взимание гонораров и финансовый анализ; календарный и документальный контроль и др. В книге довольно подробно рассматривается применение компьютеров в таких сферах, как обработка текстов (редактирование), электронная почта, информационное обеспечение управленческой деятельности применительно к юридическим службам».

В приведенном отрывке слова «юристов», «юридических» можно заменить на «служащих», «библиотекарей», «снабженцев» или любое другое из этого ряда — практически все утверждения останутся столь же верными. Специфика деятельности юридических организаций не столь велика, чтобы рекомендации для них не были верны для любой «конторы» (в данном случае это понятие используется в

смысле не обобщенно-ругательном, а просто обобщенном).

Авторы книги проделали немалую работу и выделили квинтэссенцию того, что должен узнать человек, никогда не имевший дела с компьютерами и решивший познакомиться с ними. В главе «Несколько слов о компьютерах» есть и краткий исторический очерк, и ценнейший, обычно упускаемый раздел «Чего не могут делать микрокомпьютеры»; затем читателя проводят через «Жаргонный лабиринт», заботливо держа под руку; дают ему абрис программного и аппаратного обеспечения и в заключение — несколько очевидных для специалиста, но весьма ценных для дилланта советов относительно практических действий, без которых попытка использовать ПЭВМ в работе почти наверняка потерпит крах (увы, это случается очень часто). «...В состоянии ли вы приобрести такой компьютер, который обеспечит максимально эффективное использование необходимого вам программного обеспечения? Помните, что вы ассигновали средства на приобретение системы» (курсив мой.— К. Ш.) — один только этот абзац может сэкономить уйму времени и нервов, если не забывать им руководствоваться.

Одним словом, эта книга — учебник для тех, кто не хочет изучать информатику, но намерен использовать даруемые компьютерами возможности; она же — первая хорошая ступенька для тех, кто хочет учиться, но не знает, с чего начать.

Комплиментов, пожалуй, достаточно, пора перейти к критике. Она уложится в одну фразу — именно потому, что нареканий очень

\* Микрокомпьютеры для юристов / Мак-Кланг Кр. Дж., Герриери Дж. А., Мак-Кланг К. А. м.л.: Пер. с англ. М.: Юридическая литература, 1988. 144 с.

много: неверный перевод терминов; утверждения, способные смутить не только дилетанта; подпись «Дискета» под фотографией винчестерского дисковода; а в одном месте, по-видимому, потерялся целый абзац и после подзаголовка «Совместимость» речь идет о надежности. Видимо, среди «родителей» русского текста не нашлось ни одного, достаточно знакомого с вычислительной техникой.

Впрочем, все это поправимо. Тираж книги 40 000 экз., а заслуживает она гораздо большего. Будем надеяться, что второе издание будет подготовлено более тщательно.

94

А может быть, эти огрехи не так уж и опасны? За первой книгой последует вторая, третья, и, поскольку ошибки в каждой свои, а правда — одна, путем логического умножения (функция «И») читатель найдет истину.

«Как только вы познакомитесь с удивительными возможностями микрокомпьютеров и легкостью работы с ними, вопрос "Зачем использовать микрокомпьютер?", по-видимому, заменится вопросом "Какой компьютер лучше приобрести?" — утверждают авторы. Но по-русски следовало бы сформулировать иначе, например, так: «Где достать ПЭВМ?» И поскольку доставать ее придется долго (редакция журнала занимается этим второй год), времени на чтение будет достаточно.

**К. ШЕХОВЦЕВ**

## Фантастика о настоящем



Жанр научной фантастики подчиняется вполне определенным правилам. Прежде всего фантаст описывает нечто правдоподобное, но не существующее или недоступное. Действие должно происходить либо в прошлом, как правило далеко, тогда это «фантастика о прошлом». Либо в будущем. Причем, чем оно отдаленнее, тем фантастичнее эта «фантастика о будущем». В середине временной шкалы образуется провал. Редакция научно-популярной и научно-фантастической литературы издательства «Мир» решила исправить положение и к году дракона подготовила блестящий подарок, выпустив книгу о многоруких, многоголовых и пр. под названием «Несчастье у робота профессий».\*

Во-первых, это прекрасный обзор истории и родословной роботов, их сегодняшних и будущих (научно обоснованных) применений во всех областях человеческой деятельности.

Во-вторых, это прекрасно иллюстрированный альбом, который может служить об-

разцом лучших традиций рекламы. Сто частных лиц, общественных организаций, коммерческих и системотехнических компаний предоставили для него более 240 фотографий (не считая множества прекрасно выполненных рисунков).

В-третьих, это не «серьезная научная» книга, в которой специфические термины являются ее единственным украшением. Скорее это рассказ, в котором доступность и точность определений удачно соединены в единое целое. А читатель, которому не все понятно, найдет в конце книги толковый словарь.

И наконец, это чуть-чуть и фантастика, прекрасная и интересная для всех возрастов и специальностей.

Книга рассказывает, что представляет из себя робот и что его окружает, как это механическому устройству с электронным мозгом удается заменить человека, взяв на себя монотонные операции на производстве.

Конечно, Карел Чапек не предполагал, что использованное им в пьесе «Россумские универсальные роботы» (1920) понятие «робот» (от чешского *robot* — «принудительный труд») получит столь быстрое воплощение в реальности и распространение как в научной, так и в художественной литературе. Тем не менее сегодня роботы существуют, стационарные и движущиеся; в рецензируемой книге мы встретим роботов «Бист» и «Шейки», которые напоминают своим поведением живых существ.

А за пределами предприятий для роботов нашлось столько дел, что рассказать о них можно только вкратце. Это и работа в условиях, опасных или недоступных для человека, и, например, помощь слепым, которую

\* Марш П., Александр И., Барнетт П., Дулинг Д., Гилл К., Мэтьюз П., Моравек Г. Несчастье у робота профессий / Пер. с англ.; Под ред. В. С. Гурфинкеля. М.: Мир, 1987.



оказывает японский движущийся робот-поводырь, применяя на практике три закона робототехники Айзека Азимова.

Не обижены и дети. Прежде всего из привлекают роботы-игрушки. Недорогие микророботы используются для учебных целей и развлечений, для лабораторных исследований и помощи инвалидам. Примеров в книге множество.

Но роботы уже вышли за пределы Земли, и фотографии свидетельствуют, что они могут работать на орбите и лунном грунте, как, например, советский «луноход». Космическая тема завершается обсуждением проблемы будущего — самовоспроизводства роботов. В книге обосновываются цели подобных проектов, приводятся количественные оценки необходимых для этого ресурсов и оценивается возможность выхода самовоспроизводящихся роботов из-под контроля человека. А следующая глава знакомит читателя с разработками, цель которых — наделять роботы «здоровым смыслом», т. е. с проблемами «искусственного интеллекта».

А что дальше? «Пытаясь предсказывать будущее робототехники,— говорят авторы книги,— не следует забывать, что роботы — это всего лишь часть более обширного явления — наступления эпохи информатики».

В заключение хочется сказать, что издательство «Мир» оперативно отреагировало на выход этой книги за рубежом и подготовило ее публикацию на русском языке значительно быстрее обычных сроков. Отмечая прекрасное полиграфическое исполнение, с грустью обнаруживаешь фантастическую цену — 7 руб. 50 коп. »

**А. СЕВЕРНЫЙ**

## КЛУБ ПОЛЬЗОВАТЕЛЕЙ БК

Наш журнал открывает новую рубрику. Надеемся, она окажется для вас полезной. Просим читателей присылать свои советы, разработки, предложения и отзывы на наши публикации.

## «Здравствуйте, говорит «Электроника БК-0010»!

Компьютерная речь из редкой диковины превращается в эффективное средство взаимодействия человека и ЭВМ. Мощные компьютеры, способные распознать человеческую речь, уже научились поддерживать несложный разговор. Если запас необходимых слов невелик, с разговором справляются даже мини-ЭВМ, например в кассе для продажи авиабилетов. «Электронику «БК-0010» мощным компьютером не назовешь, но и эта микро-ЭВМ в обычный диалог, когда мы нажимаем клавиши и читаем ответы с экрана, может вставить пару слов или даже фраз. Это может быть эмоциональное «Го-о-о!» в компьютерной игре «Видеоспорт», автор П. Коноплев (г. Москва) или бесстрастное «Ошибка», как сделано в некоторых учебных программах. Маленький робот на экране строит домик, рассказывая, что происходит, а говорящая программа обучает детей называть цифры; обе программы — «Домик» и «Цифры» — написаны Ю. Кузьминым из Риги. С. Ивашильников (г. Арсеньев) научил БК диктовать содержимое ячеек памяти. Работая с такой программой, при некотором навыке можно обойтись без телевизора, наверное, поэтому автор дал ей название «Слепой». Синтезировать речь можно по-разному; программа «Голос-2» москвича А. Чистякова — простейший пример. Строго говорят, эту демонстрационную речевую программу нельзя отнести к син-

тезаторам речи. БК лишь воспроизводит записанные заранее слова или фразы, подобно магнитофону. И все же с помощью этой программы можно услышать голос своего компьютера, а фрагменты программы и принципы, лежащие в ее основе еще пригодятся.

◆ ◆ ◆

Прежде чем вводить программу, нажмите клавиши ЗАГЛР пробел Т. При этом на экране появится знак +, означающий приглашение к работе в режиме «Тесты». Чтобы вводить программу, написанную в кодах, необходимо режим «Директивы отладки», поэтому нажимаем клавиши РУС Т С. Новое приглашение к диалогу больше всего похоже на букву Х с кружком посередине, на языке программистов этот символ называется черепашкой, солнышком или рублем.

Первая директива машины — команда 1000А. Так устанавливается адрес, начиная с которого в памяти БК разместится текст нашей программы. Программу вводим число за числом, разделяя числа нажатием клавиши «.». Каждое нажатие увеличивает на 2 содержимое счетчика адресов, т. е. для записи очередной команды подготавливается очередная ячейка запоминающего устройства. Ошибку можно исправить, нажав клавишу «—». В ответ на это машина выдаст на экран число из последней ячейки. Ввод правильного кода подтверждаем клавишей «.». Чтобы сверить текст программы

104006	104016	162700	000060	010037	001056	010037	001142
012700	020000	012701	040000	012703	000020	013704	177716
042704	177737	001401	000261	006005	000241	012704	000002
077401	077314	010521	077020	013700	177662	162700	000127
001402	000137	001020	013700	001022	013701	001026	012703
000020	012104	006104	000240	103003	012737	000040	177716
012705	000002	077501	012737	000000	177716	077315	077021
013700	177662	162700	000132	001402	000137	001106	000137
001000							

из журнала с кодами, которые мы записали в память, устанавливаем счетчик адресов на начальный адрес (команды 1000А) и выводим на экран содержимое ячеек (команда 220Л). Наконец программа отлажена и ее можно запустить командой 1000G. БК тотчас же запросит данные о скорости вывода речевой информации — этот параметр определяет разборчивость речи. Максимальной скорости соответствует 1, минимальной — 6. Компьютер начинает записывать информацию в экранную память; при этом данные, которые попадают в компьютер, можно контролировать не только на слух, но и визуально. Магнитофон включает в режим записи с микрофона, т. е., как усилитель. Звуковым колебаниям микрофона будут соответствовать изменения уровня сигнала на входе компьютера. Похоже, что произношение БК лучше всего тогда, когда светлые и темные пятна на экране чередуются примерно поровну. В том, что записать разборчивую речь не так уж легко, можно убедиться, если нажать клавишу W. По этой команде БК начнет читать содержимое экранной области памяти. Клавиша Z, напротив, останавливает воспроизведение. Удачную фразу записывают в основную память компьютера. Для этого нужно прервать выполнение программы («СТОП») и вновь запустить ее, изменив одну команду:

1026A2000G. Ответив на запрос о скорости, обработанную фразу начитываем в микрофон до тех пор, пока на экране не появится белая полоса — признак, что память исчерпана. Готовый «кирпичик» речи можно записать на магнитофон. Для этого нажмите клавиши «СТОП» М 3. На вопрос АДРЕС-введите 1000, на вопрос ДЛИНА-отвечайте 37000. Имя прог-

раммы — ГОЛОС-2. На магнитную ленту будет записана не только программа, но и закодированная речь.

Показывая готовую программу, ее запускают с адреса 1106. Кстати, картинка, которая получается при записи речи в экранную область памяти, по-моему, не менее интересна, чем говорящая «Электроника БК».

А. ЧИСТЯКОВ, Москва

## Программная орбита

В заметках с таким названием мы предполагаем рассказать о программных разработках для БК. Информацию о программах мы надеемся получить от их авторов, а также от тех, кто использует эти программы. Единственное условие — в адрес редакции кроме описания выслать магнитофонную кассету с записью программы. Без разрешения автора программа копироваться не будет. Возврат кассеты по почте редакция не гарантирует.

1. ОМАШКОД, автор С. Ивашинников (г. Арсеньев), машинные коды. Программа составляет основу операционной среды для БК, включающей функции редактирования текстов, отладки программ, трансляции текстов с языка МАШКОД в коды и ретрансляция машинных кодов в текстовую форму МАШКОДа. Редактируемый текст можно выводить

на экран в виде бегущей строки, причем, для того чтобы облегчить считывание текстов с экрана, символы выводятся с восьмикратным увеличением. Несмотря на продуманный сервис и довольно широкие возможности, программа МАШКОД занимает в памяти компьютера всего лишь 1,4К байта.

2. КОПИР, автор С. Ивашинников (г. Арсеньев), язык МАШКОД.

Программа написана на языке МАШКОД и является развитием известной программы СОРХ. Основное преимущество новой программы в том, что возможно копирование целыми блоками, причем можно обходиться без телевизора. Если при считывании файла возникает ошибка, БК тотчас же сигнализирует об этом, не ожидая, когда завершится считывание. Хотя текст программы содержит инструк-

цию для пользователя, программа КОПИР ориентирована на опытного пользователя БК.

3. ХУДОЖНИК, автор А. Кузьмин (г. Рига), Т-язык.

Простой графический редактор, который позволяет не только рисовать на экране, но и распечатывать полученную картинку на бумаге с помощью принтера СРА-80, входящего в комплект ПЭВМ «Агат». Принтер подключается по схеме, опубликованной в «Информатике и образовании» 1987, № 5.

4. МОРЗЕ, автор А. Кузьмин (г. Рига), Т-язык.

Прекрасный электронный помощник как для начинающего, так и для профессионального радиолюбителя. В программе содержится полный набор кодов Морзе для русского и английского алфавитов. Скорость можно регулировать от 1 до 25 слов в минуту. Текст радиogramмы вводится с помощью экранного редактора. Радиостанцию подключают к магнитофонному выходу БК.

5. РАК-88, авторы Л. Радченко, С. Кузьмичев (Москва), машинные коды.

Редактор текстов позволяет набирать тексты объемом до семи стандартных машинописных страниц, редактировать набранный текст, записывать его на магнитную ленту для длительного хранения. В отличие от других редакторов текста РАК-88 позволяет распечатывать набранный текст на отечественном принтере УВВПЧ-30-004.

6. РЕВЕРСИ, автор П. Коноплев (Москва), Фокал.

Известная логическая игра, популярная во многих странах под названием «Отелло». Цель игры — окружить своими шашками шашки противника. Аналогичная програм-

ма входит в комплект компьютера. В отличие от этой программы вариант П. Коноплева играет по всем правилам.

7. КЛАД-3, автор Д. Баранов (г. Николаев), машинные коды.

Задача игрока — выбраться из 20 лабиринтов заколдованного подземного замка, собрав при этом все клады. Нужно избежать неприятных встреч с хранителями кладов и стараться не упасть в воду. Чтобы перейти из одного лабиринта в другой, нужно не только собрать все

клады, но и разыскать ключ от закрытых дверей. Цветная графика, игра сопровождается звуковыми эффектами. Предусмотрено управление не только с клавиатуры, но и от джойстика.

8. ПЛНМ, автор Ю. Яковлев (г. Ижевск), Бейсик-Вильнюс-86.

Функцию  $y=f(x)$ , заданную 3—8 точками, можно аппроксимировать степенным полиномом (до 7-й степени). Графическое представление результатов аппроксимации можно получить на экране.

## Маленькие хитрости

◆ Независимо от исходного формата, строка X FCHR [155, 12, 154, 148, 158]; X FCHR [155, 154] устанавливает формат 32 символа в строке и убирает информационную строку в верхней части экрана.

В. Салий,  
г. Киев

на поочередной записи единиц и нулей в шестой разряд регистра внешних устройств 177716. Величина NZ определяет высоту NL — задает длительность звука. Перед обращением к подпрограмме переменным присваивают числовые значения — от 20 до 150 для NL и от 1 до 15 для NZ.

С. Комаров,  
г. Львов



◆ Собираясь работать с магнитофоном, поставьте его подальше от блока питания, чтобы помехи не помешали считывать программы и записывать их на ленту.

В. Булитко  
(ученик 7 класса),  
г. Одесса

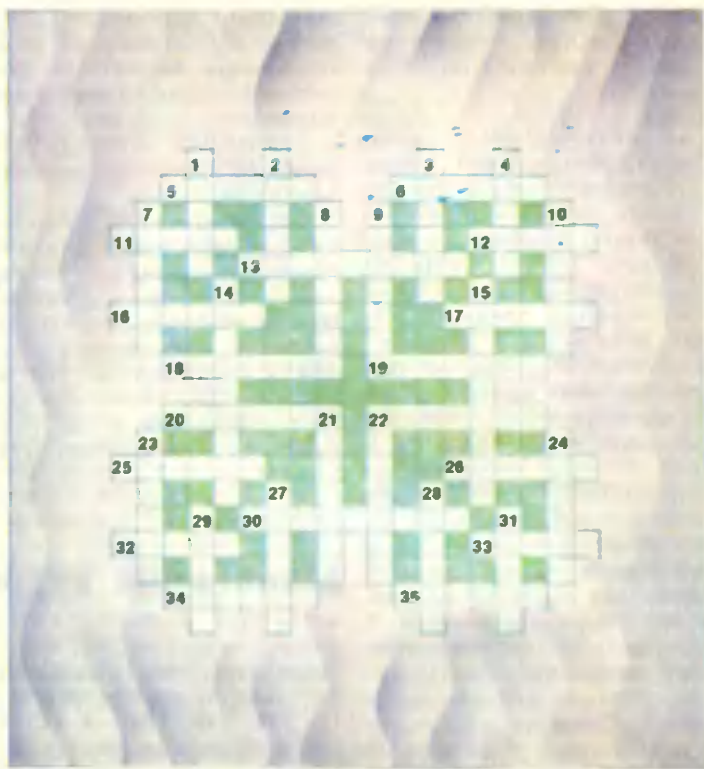
```
100 FOR N=1 TO NL
110 OUT — 50 %, 64 %, 1 %
120 FOR J=1 TO NZ
130 NEXT J
140 OUT — 50 %, 64 %, 0 %
150 NEXT N
160 RETURN
```

◆ Чтобы включить простые звуковые эффекты в программу на Бейсике-MSX, можно воспользоваться несложной подпрограммой, основанной

# Кроссворд<sup>1</sup>

По горизонтали: 5. Устройство с односторонней пропускной способностью. 6. Одна из первых американских ЭВМ. 11. Алгоритмический язык, ориентированный на решение экономических задач. 12. Несущий элемент конструкции электронной аппаратуры. 13. Формулировка алгоритма в форме, допускающей его непосредственную реализацию на ЭВМ. 16. Исследователь, предложивший один из первых алгоритмов для составления кроссвордов на ЭВМ. 17. Элемент записи идентификатора массива. 18. Алгоритмический язык, называющийся «переводчиком формул». 19. Часть света, образуемая двумя материками. 20. Последняя операция при выполнении подпрограммы. 22. Алгоритмический язык высокого уровня. 25. Широко распространенный на небольших ЭВМ язык программирования. 26. Древнегреческий философ, приговоренный к смерти и отравленный ядом цикуты. 30. Отрезок прямой, соединяющей две несмежные вершины многоугольника. 32. Несколько команд, выполняемых подряд. 33. Символьная запись программы. 34. Детский юмористический киножурнал. 35. Наука о способах доказательства и опровержения.

По вертикали: 1. Широко распространенный язык программирования для научно-технических расчетов. 2. Учебно-производственный язык программирования. 3. Язык программирования для ЭВМ средней производительности, разработанный в социалистических странах в 1967 г. 4. Средство отображения информации, используемое на стадионах. 7. Установленная форма представления данных при вводе и выводе. 8. Способ организации памяти, при котором последовательно записанная информация может быть считана только в последо-



вательности, обратной последовательности записи. 9. Устройство, позволяющее вручную вводить символ в ЭВМ. 10. Ряд символов, написанных в одну линию. 14. Системная программа, подготавливающая программу пользователя к выполнению. 15. Прибор для отображения информации об изменении какого-либо параметра. 21. Электронная схема с двумя устойчивыми состояниями, позволяющая хранить 1 бит информации. 22. Характеристика объекта, используемая для классификации или для поиска. 23. Порода собаки, участвовавшей в съемках фильма «Белый Бим Черное ухо». 24. Команда ЭВМ, используемая при работе с внешней памятью. 27. Передатчик информации.

28. Застывший кусок расплавленного металла. 29. Числовой символ. 31. Знак, отмечающий место в программе, на которое можно передать управление.

## ОТВЕТЫ НА КРОССВОРД

По горизонтали: 5. Клапан. 6. «Иллиак». 11. Кобол\*. 12. Плата\*. 13. Программа\*. 16. Мазлак. 17. Индекс\*. 18. Фортран\*. 19. Америка. 20. Возврат\*. 22. Паскаль\*. 25. Бейсик\*. 26. Сократ. 30. Диагональ. 32. Серия\*. 33. Текст\*. 34. «Ералаш»\*. 35. Логика.

По вертикали: 1. Алгол\*. 2. Рапира\*. 3. Алгамс. 4. Табло. 7. Формат\*. 8. Магазин\*. 9. Клавиша\*. 10. Строка\*. 14. Загрузчик\*. 15. Индикатор\*. 21. Триггер. 22. Признак. 23. Сеттер. 24. Запись\*. 27. Сигнал\*. 28. Слиток. 29. Цифра\*. 31. Метка\*.

<sup>1</sup> Кроссворд составлен ЭВМ по алгоритму Е. Поезда и А. Поезда. Звездочкой отмечены слова, входящие в словарь, опубликованный в журнале «Информатика и образование», 1986, № 1.

А. ГЕГЕЧКОРИ

## Проблемы обучения информатике

ЭВМ — неотъемлемый атрибут современной жизни. Однако, чем больше людей обращаются к ЭВМ, тем острее встает проблема общения «человек — компьютер»: использование этих сложных устройств требует определенной квалификации и опыта.

Таким образом, неспециалисту, который хотел бы воспользоваться услугами ЭВМ, приходится обращаться к посреднику либо ставить и программировать свою задачу лично.

Если в 60-х гг. первый путь решения проблемы представлялся очевидным и разумным, то сегодня, в связи с бурным расширением сферы использования ЭВМ, столько специалистов взять, в принципе, неоткуда.

А в условиях, когда к услугам ЭВМ сплошь и рядом прибегают непрофессионалы, вопрос общения человека с компьютером становится особенно актуальным.

Суть проблемы состоит в том, что человек с его гибким умом, неформальным поведением и мышлением отделяет от ЭВМ с ее «железными мозгами» пропасть. А языки программирования не мост над ней, а скорее островок, сооруженный на пути.

Ясно, что сокращать пропасть между человеком и ЭВМ можно двумя путями. Первый из них состоит в приближении ЭВМ к человеку, т. е. в разработке средств, облегчающих пользователю процесс реализации своей задачи на ЭВМ. Второй, соответственно, предполагает приближение человека к ЭВМ, т. е. обучение каждого потенциального пользователя основам программирования\*, «компьютерной грамотности».

\* Исторически сложилось так, что под программированием часто понимают не весь этап ре-

К сожалению, ни в том, ни в другом сколь-либо значительных успехов пока не достигнуто.

Средства автоматизации программирования, по-видимому, можно разрабатывать бесконечно. Это позволит ставить и решать новые, ранее недоступные задачи, затрачивая на них все меньше сил и средств, и, хотя за последние сорок лет разработчики программного обеспечения прошли большой путь от машинных кодов до таких языков, как Алгол-68 и Ада, проблема общения с компьютером по-прежнему стоит остро и средства общения с ним остаются желать лучшего. Образно выражаясь, если от машинных кодов до языков высокого уровня — километры, то расстояние до человека измеряется парсеками.

И хотя ликвидация пропасти путем приближения ЭВМ к человеку весьма заманчива, она связана с большими трудностями и затратами. Возможно, именно поэтому в последнее время упор делается на другой путь. Ясно, что человек — гораздо более гибкая система и «приблизить» его к машине гораздо «легче», чем «тащить» машину к

шения задачи на ЭВМ от формулирования целей проекта до тестирования программ, а лишь его часть, включающую кодирование и отладку. Некоторые специалисты, зная, насколько важны предшествующие кодированию этапы, считают их неотъемлемой частью процесса, именуемого программированием. В этом ракурсе понятие «программирование» приближается к понятию «информатика» — как науке об обработке информации с помощью ЭВМ. Именно в таком смысле мы будем употреблять это понятие в дальнейшем.

человеку. Основам информатики и вычислительной техники обучают сегодня уже в школе, однако и этот вариант наталкивается на определенные трудности.

Некоторые из них, такие, как отсутствие достаточного количества средств ВТ и квалифицированных преподавательских кадров, могут быть преодолены в ближайшем будущем. Решение же других проблем связано со сложными научно-исследовательскими задачами, над которыми, по-видимому, предстоит много работать ученым разных направлений.

Одна из таких задач — обучение человека программированию. Споры ведутся обо всех аспектах этой большой проблемы, нет единого мнения не только о том, чему и как надо учить, но и о том, чему и как учить не надо.

Э. Дейкстра [1] пишет, что ему «встречалось много учебников по вождению автомашин, в которых человека учат, как ухаживать за машиной, вместо того чтобы научить его, как использовать машину, чтобы попасть в нужное ему место».

Похожую мысль высказывает и Ю. Баяковский [6]. Он отмечает, что в литературе (как зарубежной, так и отечественной) получила распространение следующая схема: обширный раздел посвящается системам счисления, затем излагаются приемы программирования на машинном языке с привлечением символических обозначений. В заключение изучается какой-либо язык программирования. Типичным учебником такого рода может служить пособие [7]. По похожей схеме построена и книга американских специалистов [8]. Вначале в ней приводятся сведения о различных системах счисления, в частности китайской,

римской, египетской, вавилонской, греческой и других, после чего вводится понятие двоичной системы счисления, далее следует описание составных частей различных устройств ЭВМ и даются сведения об организации данных. Вслед за этим описываются различные области применения компьютеров, а процессу алгоритмизации и решения задач с помощью ЭВМ внимания почти не уделено.

Недостатки такой схемы обучения критикуются многими специалистами [1], [2], [6], [9]. По сути все замечания сводятся к мысли о том, что человек с самого начала сталкивается с особенностями конкретной ЭВМ или спецификой данного языка программирования, в результате чего упускается из виду концептуальная целостность предмета. Иными словами, увязнув в деталях, обучаемый, как говорится, «за деревьями леса не видит».

А между тем вопрос о том, какими будут первые понятия обучаемого программированию и как он их усвоит, представляется чрезвычайно важным, ибо это накладывает отпечаток на весь дальнейший «образ мышления» при постановке и программировании задач\*.

Очень важно, чтобы на первом этапе обучения «мощные средства» выбранного языка программирования не заслоняли собой процесса олицетворения алгоритма. Следует помнить, что программа никогда не является самоцелью; она предназначена для того, чтобы вызвать преобразование инфор-

\* Не потому ли Э. Дейкстра не любит обучать студентов, которые на младших курсах изучали Фортран [10]?



мации, которое в свою очередь нужно для получения того или иного результата [1].

По мнению некоторых специалистов, в вводном курсе программирования «должны рассматриваться вопросы систематического конструирования алгоритмов, а сам курс должен быть составной частью общего математического образования» [6].

Существует также мнение, что общению с компьютером надо обучать с детства. Интересен опыт, накопленный английскими и болгарскими специалистами [12], [13].

Сторонники такого подхода придерживаются мнения, что понятия, которыми при составлении программы оперирует человек, обученный программированию с детства, являются для него естественными. Важно лишь выбрать подходящую форму для их выражения. Большое значение придается непосредственно общению с компьютером, игровым упражнениям, мастерски подобранным и интерпретированным [12].

Отсутствие единого взгляда на процесс обучения программированию является, по-видимому, не случайным и имеет в своей основе ряд причин.

Одна из них состоит в том, что программирование — очень молодая наука, многие понятия которой еще только формируются. Вследствие этого вопрос о том, что в ней фундаментально важно, а что является собой надстройкой, не очевиден. «В информатике еще не было своего Галилея и Ньютона... Однако прежде всего здесь нужен Уильям Оккам, который сказал, что существностей не следует приумножать без необходимости» [14].

Возможно, именно поэтому вызывает споры вопрос о том, что именно следует включить в программу обучения. Действительно, мало у кого вызывает сомнение, что можно программировать на ЭВМ задачи, не зная, например, систем счисления, или не имея понятия о ферромагнитных сердечниках и триггерах, или не зная Фортрана. Можно даже не знать машинных команд, иметь смутное представление о каналах ввода-вывода и вообще не иметь представления об ОС/360 и ПЛ/1.

Тем не менее вышеперечисленные понятия в той или иной совокупности занимают едва ли не большую часть многих курсов по информатике.

Дело в том, что многие отождествляют трудность программирования с трудностью изощренного использования громоздких и причудливых сооружений, известных под названием «языки программирования высокого уровня» или — еще хуже! — «системы программирования» [2].

Между тем это трудности совершенно разного рода. «Понимать, что такое глина,— не значит понимать, что такое горшок» [14]. Можно быть грамотным человеком и никудыщим писателем.

Можно знать Кобол, Фортран, ПЛ/1, операционную систему и споткнуться при программировании какой-либо нетривиальной задачи.

Язык программирования — это всего лишь инструмент, с помощью которого достигается цель, причем целью является не программа, а те преобразования информации, которые этой программой выполняются.

Мы по сей день встречаемся с отождествлением обучения программированию с обучением языкам программирования. Организация СНПО «Алгоритм», претендующая у нас в стране на центральную роль в обучении профессиональных программистов, ведет три курса в программировании. Один из них включает изучение ПЛ, другой — Фортрана, третий — ассемблера. Внимание слушателя при этом целиком поглощено подробным ознакомлением с характерными особенностями различных языков и он начинает думать, что чем больше таких особенностей он поймет, тем лучшим программистом он станет, ему и невдомек, что все эти «мощные характеристики» в большей мере принадлежат к множеству проблем, чем к множеству решений» [9]. При таком подходе обучаемый в лучшем случае станет искусным кодировщиком тривиальных алгоритмов, и трудности, с которыми он столкнется, никогда не будут связаны с самой проблемой, а всегда с заданным способом кодирования алгоритма, «после нескольких таких курсов по-настоящему трудные задачи останутся за пределами его умственного кругозора» [9].

Есть у этого мнения и противники, проповедующие, что при обучении надо уделять первостепенное внимание непосредственно тем вещам, которые понадобятся на производстве, в жизни, пусть даже это будут такие детали, как, например, устройство винчестеровских дисководов [1]. Подобный подход преследует весьма утилитарные цели: научить пользователя программировать задачу и решать ее на конкретной машине. И если в каком-то частном случае этот подход можно считать оправданным, то возводить его в догму при преподавании программирования вообще вряд ли целесообразно. При подготовке же профессиональных программистов такой подход в принципе неприемлем.

Другая причина, провоцирующая расхождение во взглядах на процесс обучения программированию, состоит в следующем.

Математическое обеспечение и электроника сделали за последнее время большой шаг вперед — от электронных ламп до микропроцессоров и от машинных кодов до таких языков, как Пролог и Ада. В то же время архитектура ЭВМ за последние сорок лет осталась практически на том же уровне, если сравнить архитектуру большинства наиболее распространенных современных ЭВМ (IBM/360 и IBM/370, PDP-11, VAX-11), с архитектурой первых ЭВМ с запоминаемой программой, то окажется, что появление всех существенных различий датируется 50-ми гг. [4].

В результате этого возник феномен, известный под названием «семантический разрыв». Сущность его состоит в том, что объекты манипулирования и соответствующие им операции, реализуемые архитектурой вычислительной системы, редко имеют близкое родство с объектами и операциями, реализуемыми в языках программирования. В свою очередь объекты и операции языка программирования зачастую очень далеки от понятий, которыми оперирует пользователь при решении своей задачи.

Рассмотрим чуть подробнее некоторые аспекты этого разрыва на примере одной из самых популярных в мире архитектур — системы IBM/370.

В архитектуре этой системы отсутствуют средства для обработки структур и строчковых данных; специальные средства для организации процедур блочной структуры; представление некоторых видов данных, имеющихся в языках программирования; нет специальных аппаратных средств, ориентированных на реализацию таких важных принципов, как модульность, сокрытие информации, использование абстрактных типов данных и мониторов.

Отметим также отсутствие единообразного подхода к организации памяти. Запоминающая среда каждого типа имеет свою систему адресации, распределения памяти и ее защиты. Причем алгоритмы доступа к данным различны даже среди однотипных устройств (так, например, для 7-мегабайтных дисков используется один алгоритм доступа, а для 29-мегабайтных — другой); о пропасти между оперативными и внешними запоминающими устройствами мы уж и не говорим.

Следствием этого «семантического разрыва», этой «оторванности» средств решения от самой задачи является одно из главных затруднений при программировании. «Довольно трудно объяснить что-либо (например, лингвистические концепции) в терминах примитивного словарного запаса (т. е. машинными кодами), не имеющего ничего об-

щего с тем, что подлежит объяснению...» [16].

По-видимому, именно эта пропасть между объектами и операциями вычислительной машины и понятиями, которыми оперирует пользователь, и провоцирует перегруженность многих курсов программирования всевозможными техническими деталями.

Углублению упомянутого «семантического разрыва» способствует и чрезмерная «математизация языков программирования» [17]; поскольку первыми пользователями и разработчиками ЭВМ были математики, математический подход к языкам программирования нашел отражение в одном из первых языков — Фортране. И хотя с тех пор сфера применения ЭВМ значительно расширилась и возникло даже целое направление, которое, в отличие от традиционного программирования вычислительного характера, получило название нечисленного программирования (обработка символьной информации, системное программирование и т. д.), математический, «формульный», подход прослеживается во всех последующих языках, вплоть до недавно разработанных, таких, как Ада и Пролог.

Между тем быть программистом вовсе не означает быть математиком [17]. Тот факт, что программисты (нематематики) в своей нематематической работе пользуются прежде всего математическим оружием, определяется не тем, что это оружие их удовлетворяет, а скорее тем, что у них просто нет другого.

Несоответствие между программистской деятельностью и «инструментом», используемым в этой деятельности, приводит к тому, что целый ряд понятий вводится неестественно, не так, как они должны бы мыслиться пользователем, а так, как их удобнее использовать с помощью «инструмента», будь то архитектура ЭВМ или математический аппарат.

Рассмотрим некоторые из этих понятий. **Идентификатор.**

Под идентификатором мы понимаем имя, что-либо обозначающее.

Мы привыкли к тому, что сначала должен существовать объект и только потом его имя. Не может существовать имя, которое ничего не обозначает. (Этой ситуации в естественном языке соответствует бесмысленный набор звуков.) В то же время в языках программирования сплошь и рядом сначала описывается имя, а затем ему присваивается (а может, и нет) какое-то значение. Это не корректно с конструктивной точки зрения — «сначала объект должен быть создан, лишь затем поименован» [17].



## Типы данных.

Довольно неестественным с точки зрения пользователя является разделение арифметических данных на два типа — целочисленный (Integer) и действительный (Real), принятое, за редким исключением (например, Кобол), во всех языках. Ведь по логике вещей действительные числа включают в себя как целые, так и дробные, так зачем же их разделять? И уж совсем затруднительно объяснить (а равно и понять), почему число 25 относится к целому типу, а 25.0 — к вещественному, ведь это одно и то же!

Объясняется этот «феномен» просто. Разделение арифметических данных на типы вышеупомянутым способом происходило исходя не из нужд и понятий пользователя, а из внутреннего представления данных в ЭВМ. Целому типу соответствует представление данных с фиксированной, а действительному — с плавающей точкой; целые и нецелые числа тут ни при чем.

## Переменные и константы.

Подобный, аппаратный подход обуславливает, по-видимому, и не совсем корректное деление данных на константы и переменные. Во многих языках отсутствует строго определенное деление данных на константы и переменные, аппаратно же это деление вообще никак не поддерживается. Не потому ли в некоторых учебниках константа определяется чуть ли не как разновидность переменной, которая не меняет своего значения [8]? А ведь это совершенно разные вещи. Константа не может изменить своего значения в принципе; переменная же, если она не изменила своего значения в процессе выполнения программы, не стала от этого константой.

## Ввод-вывод и запоминание-вспоминание.

Прежде чем информацию обработать, ее нужно получить. Следовательно, каждая система переработки информации обладает средством ее ввода. После того как информация введена, система осуществляет ее «восприятие», выражаясь языком программистов, происходит выявление согласованности введенной информации с заданными спецификациями. Будучи введенной, информация может быть обработана и при необходимости «запомнена». Если спустя какое-то время нам вновь понадобится эта же информация, то ее уже не надо вводить вновь, мы можем ее «вспомнить». Очевидно, что и «ввод» и «вспоминание» — это не одно и то же и что подобные операции присущи любой системе обработки информации, в том числе и ЭВМ. Тем не менее грань между этими разными понятиями никак не проводится. Обе эти операции именуются вводом (выводом)

информации и реализуются одними и теми же операторами. Опять налицо давление «инструмента». С точки зрения техники «ввод-вывод» и «запоминание-вспоминание» — идентичные операции, представляющие собой передачу информации от оперативной памяти к внешнему устройству или, наоборот, от внешнего устройства к оперативной памяти. Наверное, именно поэтому эти понятия не различаются синтаксически в языках программирования, а уж как следствие — не различаются в описаниях и инструкциях.

## Внешняя память и внешние устройства.

Одна неточность порождает другую. Зачастую в литературе по вычислительной технике мы можем прочитать, что внешняя память ЭВМ — это накопители на МЛ, МД или БД [3]. Подобное утверждение содержит одну маленькую, но принципиальную неточность.

Очевидно, что любая система обработки информации должна обладать, как минимум, двумя типами памяти — оперативной, где находится информация, которой система оперирует в данный момент времени, и долговременной, в которой хранится все остальное. Эту вторую, долговременную, память обычно именуют внешней во время рассмотрения ситуации на уровне понятия «вспоминание», внешняя память является логическим понятием и не имеет значения, как она реализована, важно, что она есть.

Если же посмотреть на этот вопрос со стороны техники, то накопители на магнитных носителях представляют собой устройства, для которых исторически сложилось название «внешняя память». Но это уже понятие физического уровня, и его не следует путать с предыдущим понятием — «долговременная память».

Таким образом, накопители на магнитных носителях — это не долговременная память, а устройства, которые могут использоваться в качестве долговременной памяти. Это не мешает им в другом случае выступать в качестве устройств ввода-вывода, как это и происходит, например, при подготовке данных на магнитной ленте.

Теперь, когда мы подробно рассмотрели «пропасть», разделяющую человека и ЭВМ, нетрудно видеть, как через нее «наводятся мосты». По сути дела, все возложено на программиста, в помощь которому придана операционная система.

Учитывая приведенные факты и соображения, становится более ясным, почему программирование по сей день напоминает «асфальтовую топь, в которой увязли многие большие и сильные звери» [5]. Почему программа, свободная от ошибок, представляет собой «абстрактное теоретическое понятие»,



почему ощущается дефицит четкой методики обучения программированию.

В чем же видится выход?

Существует мнение, разделяемое многими специалистами, о том, что самый важный язык, который следует знать программисту, — это не ПЛ/1, а родной язык [15]. Иными словами, самое главное — научиться ясно и просто излагать свои мысли на любом языке, а уж потом, имея ясно сформулированную мысль, искать способы ее воплощения в конструкциях языка алгоритмического.

А ведь чаще всего бывает наоборот, когда обучаемый, владея несколькими популярными конструкциями, пытается наложить их на еще «сырой» алгоритм.

И это потому, что зачастую единственная цель, которая стоит перед обучаемым, — результат\*. Неважно, как он будет получен, чем быстрее, тем лучше; победителей не судят.

Такая точка зрения справедлива для чего угодно, только не для процесса обучения. Тут, по нашему мнению, важен процесс получения результата; именно на этом должны быть сосредоточены усилия преподавателя.

#### Литература

1. Дал У., Дейкстра Э., Хоар К. Структурное

\* Еще раз напомним, что под результатом следует понимать те вычисления, которые выдает программа.

программирование. М.: Мир, 1975.

2. Дейкстра Э. Дисциплина программирования. М.: Мир, 1980.

3. Криницкий Н. А. Алгоритмы вокруг нас. М.: Наука, 1984.

4. Майерс Г. Архитектура современных ЭВМ. М.: Мир, 1985.

5. Брукс Ф. П. Как проектируются и создаются программные комплексы. М.: Наука, 1979.

6. Вирт Н. Систематическое программирование. М.: Мир, 1977.

7. Жоголев Е. А., Трифонов Н. П. Курс программирования. М.: Наука, 1971.

8. Reachellie S. Heller, C. Dianne Martin. Bits'n Rytes about Computing; A Computer Literacy Primer. University of Maryland, 1982.

9. Dijkstra E. W. On the teaching of Programming. I. E. On the teaching of Thinning. Lect. notes Comput. Sci. 1976.

10. Иодан Э. Структурное проектирование и конструирование программ. М.: Мир, 1979.

11. Your computer. Sept. 1985. P. 59.

12. Sharon, Boren. An apple for the kids. 1984.

13. Николов Р. Лого първи клас

14. Кей А. Программное обеспечение ЭВМ // В мире науки. 1984, ноябрь.

15. Лингер Р., Миллс Х., Уитт Е. Теория и практика структурного программирования. М.: Мир, 1982.

16. Weizenbaum J. Computer Power and Numan Reason; from Judgment to Calculation: San Francisco Freeman. 1976.

17. Велишвили Н. М., Доквадзе К. А. О природе языков программирования // Психологические проблемы создания и использования ЭВМ. М.: МГУ, 1985.

18. Ершов А. П., Монохов В. М. Основы информатики и вычислительной техники. М.: Просвещение, 1985.

## Чего не хватает уроку информатики

или о том, что проблемы существуют не только при безмашинном обучении

Уже второй год учащиеся 42-й тбилисской ФМШ проводят часть уроков информатики в дисплейном классе ГВЦ Минпроса ГССР, где расположен КУВТ «Ямаха». Ученики разрабатывают на Бейсике самые разнообразные программы: графические, расчетные, игровые, информационные. Однако и проблемы в ходе обучения возникают разнообразные.

Поскольку в настоящей статье учтены мнения и предложения учеников, автор выражает благодарность своим учащимся 1986/87 учебного года за терпение и сотрудничество.

Прежде чем перейти к проблемам, необходимо сформулировать представление о целях обучения, порождающих эти проблемы.

Цель «минимум» состоит в том, чтобы заинтересовать учеников компьютером, сделать его знакомым и привычным. Компьютер должен ассоциироваться с положительными эмоциями.

Цель «миди», достижимая при 2—3 часах информатики в неделю и наличии ЭВМ, — подготовить человека, способного самостоятельно написать и отладить программу размером в 200—300 строк, если в ней не содержится особо сложных алгоритмов (рекурсии, логики с более чем 5—6 проверяемыми условиями и т. п.) и сложных структур данных (деревьев, стеков и т. п.).

Есть предложения готовить в курсе информатики людей, умеющих пользоваться программами, но не умеющих программировать. На наш взгляд, для этого не стоит вводить новый предмет. Информатика — наука об обработке информации, в частности об алгоритмизации и программировании. Работать с готовыми программами целесообразно при изучении других предметов.

Цель «максимум» — подготовить человека, уверенного в своей способности решать задачи с помощью компьютера, умеющего передавать свои знания другим, умеющего формулировать возникающие вопросы в терминах информатики.

Если же рассматривать не отдельных учеников, а школьную информатику в целом, то в качестве цели «максимум» следует выдвигать необходимость участия школьников в разработке не только учебных, но и производственных задач. Уже при 3 часах информатики в неделю за год можно подго-

товить учеников к работе, которая будет представлять практическую ценность. На эту возможность и необходимость работает несколько факторов:

прикладной, инженерный характер информатики (точнее, той ее части, которая преподается школьникам);

значительный интерес, который вызывают у учеников компьютеры;

стоимость и возможности техники, которая размещена и будет размещаться в школах.

Если сельской школе выделяют работающий комбайн, то обученных школьников, надо полагать, не заставляют косить «учебную» траву, а предлагают им работать на настоящем поле.

Рассмотрим проблемы, связанные с техническим оснащением информатики.

Сегодня каждый урок на ПЭВМ начинается и кончается «почтовыми операциями». Каждому ученику нужно прочитать с диска и передать его программу, а в конце урока записать ее обратно. То и другое занимает минут по 15. Следовательно, ученик теряет треть урока, а учитель — две трети. Вторая проблема связана с наличием одного дисковода в классе. Скоро появятся программы, которые работают с диском (у нас это программа составления школьного расписания и ведения журнала, профориентационного тестирования, игротехи). Значительную часть урока учащиеся, разработчики таких программ, простаивают. Первую из этих проблем можно решить созданием сетевых процедур рассылки и сборки, когда пересылка программ между диском и дисплеями идет автоматически, по именам, задаваемым на ученических дисплеях. Инициироваться эти процедуры должны с учительского компьютера. При этом необходимо установить правила построения уникальных имен программ для всех классов и учеников.

Решить обе проблемы можно, установив на каждые 3—5 ученических ЭВМ по дисководу.

Все, что сказано о дисководах, относится, хотя и в меньшей степени, к принтерам.

Если в ходе урока произойдет остановка ВТ, то урок неизбежно срывается — перестроить его на «теоретический» очень трудно и учителю, и ученикам. При имеющемся

уровне надежности отечественной техники «камнем преткновения» становятся нерезервированные компоненты дисплейного класса: учительский дисплей, сеть, дисковод и принтер.

Трудно переоценить действие на детей таких возможностей компьютера, как графика, цвет и звук. Яркие зрительные образы, рождая начальный интерес, служат впоследствии наградой за проделанную работу. А выход на музыкальные программы сломил сопротивление информатике самых заедлых гуманитариев. Поэтому школьный компьютер немислим без этих возможностей.

В школе профессиональных программистов почти нет. Тем более странно, что вычислительная техника поставляется в школы практически «голышом» — без программного обеспечения. На первых порах достаточно было бы собрать то, что имеется (в частности, о созданном программном обеспечении писалось на страницах журнала «Информатика и образование»), и поставлять эти программы вместе с компьютерами. Единственное, что хотелось бы добавить в эту библиотеку, — набор «образцовых» программ, т. е. хорошо откомментированных, работающих программ, в которых предусматривалась бы работа с графикой, звуком, спрайтами, прерываниями и т. п. На основании таких образцов легче начать делать свои программы.

Для того чтобы квалифицированно вести урок, одних только компьютеров и программ недостаточно. Необходимы наглядные пособия: плакаты и руководства. Хорошо, если бы КУВТ поставлялся вместе с другим оборудованием кабинета. Правда, об облицовке стен и электрооборудовании пока мечтать нескромно, но необходимо иметь хотя бы комплект плакатов по технике безопасности.

К машине в достаточном количестве должны прилагаться руководства. По школьному учебнику с машиной не поработаешь, а профессиональной литературы в магазинах не хватает и специалистам.

Создавать или закупать для школ «игрушечные», пригодные только для обучения, компьютеры — неоправданная расточительность по отношению к творческому потенциалу учеников. Уже при 2—3 часах информатики в неделю ученики способны на втором (а некоторые и на первом) году обучения создавать программы, представляющие не только учебную ценность. Сейчас участие школьников обучающихся на «Ямахе» или КУВТ-86, в разработке программ для различных организаций исключается только по той причине, что аналогичной техники в организациях нет и быть не может. Тот факт, что большинство компью-

теров «знает» Бейсик, ничего не решает. Сегодня вариантов Бейсика примерно столько же, сколько типов ЭВМ. Поэтому школьные компьютеры должны быть как минимум совместимы снизу вверх с основными рядами профессиональных компьютеров по программному обеспечению и, безусловно, должны быть совместимы по носителям информации и интерфейсам.

Быть может, наилучший выход состоит в том, чтобы развиваемые сегодня ряды «Электроник», «Искр» и т. д. имели бы младшего (и любимого) брата — школьного компьютера.

Остановимся на проблемах, связанных с методикой.

В театральных кругах известно, что «даже гениальный актер не может соперничать с открывающейся дверью». Может ли учитель, особенно если он не гениальный актер, соперничать с окном в новый и увлекательный мир, распахнутый перед каждым учеником, сидящим за ПЭВМ? Попытка объяснить новый материал при работающих дисплеях гарантированно срывается. Наивно полагать, что все 15 учеников нажмут именно те клавиши, которые предлагает учитель. В результате компьютеры могут попасть в различные состояния, часть из которых неизвестна и учителю. Для решения этой проблемы мы пробуем применять прием «заведенная пружина». Это означает, что на уроке, предшествующем выходу на машину, ученикам дается примерный план работ и текст программы-образца, с которого нужно начинать разбираться в новой теме. Когда ученикам известно генеральное направление урока и есть программа-образец, разброс в состоянии дисплеев оказывается значительно меньшим.

Банка задач, который столетиями накапливался в математике, физике, химии, учителя информатики лишены. Два десятка алгоритмов, кочующих из одного учебного пособия в другое, кончаются очень быстро, а главное, они совершенно не затрагивают увлекательных свойств современных компьютеров: графики, спрайтов, прерываний, музыки. Кроме задач необходимы банки заданий на разработку программ, оставляющие место для фантазии. Например, возможны такие задания: «Сделать на экране новогоднюю открытку (мультик)», «Сделать эмблему (рекламу) школы с музыкальным сопровождением». Третий, еще более сложный, но посильный сегодняшним школьникам уровень работ — создание программ и систем по техническим заданиям. В нашей практике это программы составления школьного расписания, ведения школьного журнала, профориентационного тестирования. Вопрос

о том, кто и что будет конкретно делать, нужно решать с учетом наклонностей учеников.

При наличии широкого спектра задач и технических заданий можно поддерживать на высоком уровне заинтересованность учеников в работе с ЭВМ.

В заключение сформулируем некоторые предложения:

архитектура КУВТ должна быть такова, чтобы в ней не возникало узких мест по надежности и производительности;

КУВТ должен обладать всеми современными возможностями: цветом, графикой, звуком, мультимедиа и пр. Он должен представлять оснащенный программным обеспечением, документацией в достаточном количестве и всем необходимым для работы

в дисплейном классе;

школьные компьютеры должны быть совместимы с профессиональными;

целесообразно использовать в преподавании информатики методику В. Ф. Шаталова; автор предлагает свою помощь всем, кто готов идти по этому пути;

представляется желательным создание «Задачника ИНФО» и «Фонда алгоритмов и программ ИНФО», в последнем печатались бы «образцовые» программы;

необходимо создать не один, а целый спектр учебников для различных ситуаций обучения;

желательно предоставить учителям информатики возможность очно или заочно обмениваться своими «маленькими методическими хитростями».

**М. КУШНИР,**  
учитель информатики, Москва

107

## Компьютеризацию начали. Что впереди?

Компьютеризация началась и породила море проблем. Среди них можно выделить как перспективные, так и требующие безотлагательного решения. В своих заметках я хочу поделиться предложениями относительно проблем сегодняшних.

Наиболее остро стоит вопрос о ремонте ВТ. Для его решения считаю необходимым выделить для кабинетов ВТ ставку инженера-лаборанта и обеспечить ПЭВМ минимальным набором запчастей. Инженер-лаборант нужен для проведения мелкого текущего ремонта и помощи учителям-предметникам на компьютерных уроках. В нашей школе такой опыт имеется, но основан он на энтузиазме «компьютерщиков». В ближайшем будущем наличие такой ставки должно стать нормой, и на энтузиазм здесь рассчитывать не приходится.

Тот, кто пытался не купить — оставить открытки на литературу по ВТ, столкнулся с другой проблемой: ее дефицитом. Убежден, что для преподавателей ОИВТ снабжение такой литературой должно осуществляться централизованно. Специальная литература по ВТ и программированию необходима для повышения квалификации учителей и развития увлекающихся ребят.

Срочно нужно решать проблему внешних носителей информации. Сегодня большинство ПЭВМ работают с гибкими магнитными дисками, ресурс работы которых — 1—2 го-

да, а достать их... Кроме того, в нашей школе вынуждены списывать программы с экрана вручную. Деньги есть, а достать принтер не можем!

Последнее время предпринимаются попытки объединить усилия учителей информатики, однако пока они не приносят результата. Почему? Мне кажется, что объединению препятствуют три фактора: во-первых, недостаток квалификации в профессиональном плане и нехватка наработанного серьезного багажа, необходимого для взаимопользования обмена; во-вторых, несовместимость ВТ (в этом случае объединение возможно на почве методики, но она еще не достаточно разработана); в-третьих, разноязычие: многообразие несовместимых диалектов Бейсика, традиционные Фортран, Алгол, Паскаль, русскоязычные системы.

Мое мнение: если мы хотим в результате координации наших действий наработать банк обучающих, развивающих, управляющих и других программ, мы должны обеспечить языковую и техническую базы, для того чтобы добиться совместимости ППС.

В качестве языковой базы мы должны взять язык, отвечающий современным требованиям программирования, имеющий для различных версий широкое стандартное подмножество, не очень сложный и достаточно широко используемый. Этим характеристикам, на мой взгляд, лучше всего отве-

чает Паскаль, созданный специально для учебных целей, но в силу изящности и достаточной мощности нашедший гораздо более широкое применение. Его минимальный объем, описанный в брошюре С. А. Абрамова, Е. В. Зимы «Начала программирования на языке Паскаль» (М.: Наука, 1987), практически соответствует алгоритмическому языку учебника по информатике. И не надо бояться его англоязычности.

Для обеспечения совместимости существующего «зоопарка» ЭВМ необходимо, на мой взгляд, выделить базовую модель ПЭВМ, а затем разработать для нее набор интерфейсов со всеми существующими и ожидаемыми в школах моделями. Тогда можно будет поставить в каждую школу 1—2 ЭВМ базовой модели, а остальные связать с ними. Это обеспечит задел для создания информационной сети школ.

И наконец, о проблеме, с которой пока сталкиваются не все. Это создание учащимися программных продуктов.

Сейчас для организации производительного труда приветствуются трудовые договоры. Ребятам, обучающимся по профессии «оператор ЭВМ» с «креном» в программирование, при наличии квалифицированного преподавателя по плечу разрабатывать довольно серьезные программные продукты. Технология их создания традиционна: преподаватель структурирует задачу и распределяет получившиеся логически завершенные подзадачи между ребятами с учетом их интересов и квалификации, постоянно куриру-

ет их реализацию, так как учащиеся не всегда удачно структурируют свои подзадачи, а потом осуществляет синтез полученных готовых форм. Польза от такой работы обоюдная: квалификация преподавателя растет, так как он создает идеологию всей задачи, а ребята развиваются благодаря тому, что на конкретной задаче получают предметные навыки и видят, что огромные, сложные и интересные системы состоят из кирпичиков, которые они сами в состоянии создать.

На этом пути мы сталкиваемся с проблемой заказчика и оплаты. В настоящее время оплату можно осуществить через центры НТТМ, но если есть заказчик. Правда, для этого желательно внести дополнение к положению о временном молодежном творческом коллективе (ВМТК), предусматривающее возможность создания ВМТК из школьников. Можно ожидать, что сейчас, в пору экспериментов, нам пойдут навстречу.

Еще труднее для школьников с заказчиком. Безусловно, ребятам ближе всего проблемы, связанные с учебным процессом, однако в этой области заказчика нет. Надеемся, что пока. У нас, например, близка к завершению интересная разработка: файловая система «электронный задачник», позволяющая строить любые задачки с текстом, данными и рисунками на ПЭВМ «Правец-82». Вот если бы заказчиком выступил Госкомитет СССР по народному образованию. Но остается проблема совместимости!

Итак, компьютеризация началась...



**С. МАРКОВ,**

доцент кафедры программно-методического обеспечения ЭВМ ВИПК РР и С ПТО

## Компьютеризированная система непрерывного повышения квалификации работников профтехобразования

Существующая в настоящее время организация системы повышения квалификации не удовлетворяет современным требованиям. Одним из направлений ее совершенствования, принятом в постановлении партии и правительства «О перестройке системы повышения квалификации и переподготовке руководящих работников и специалистов народного хозяйства», является переход от периодического к непрерывному обучению. В решении этой задачи важная роль отводится ЭВМ. Компьютеризация системы повышения квалификации приводит к качественным изменениям во всех ее структурно-функциональных элементах. Возможность осуществлять обучение практически без участия преподавателя расширяет понятие процесса повышения квалификации по времени и месту, дает возможность включить в него как виды обучения в системе повышения квалификации, так и трудовую и общественную деятельность, которая, по словам К. Маркса и Ф. Энгельса, есть «...главная область приложения, а стало быть, и дальнейшего развития основных существенных сил человека, его личностного потенциала» [1].

Анализируя виды обучения в системе повышения квалификации, обеспечивающие его непрерывность, можно выделить те из них, которые могут быть автоматизированы. Для каждого вида определяются соответствующие программные обучающие средства:

педагогические программные средства (ППС) повышения квалификации для самообразования;

ППС повышения квалификации и переподготовки для ИПК;

компьютерные деловые игры (КДИ) для индивидуального и коллективного обучения в СПТУ, методкабинетах и ИПК;

экспертные и справочные системы (ЭСС) (юридические, психологические, педагогические и т. п.), оказывающие помощь в решении сложных вопросов практической деятельности училища и обучающие современным методам работы.

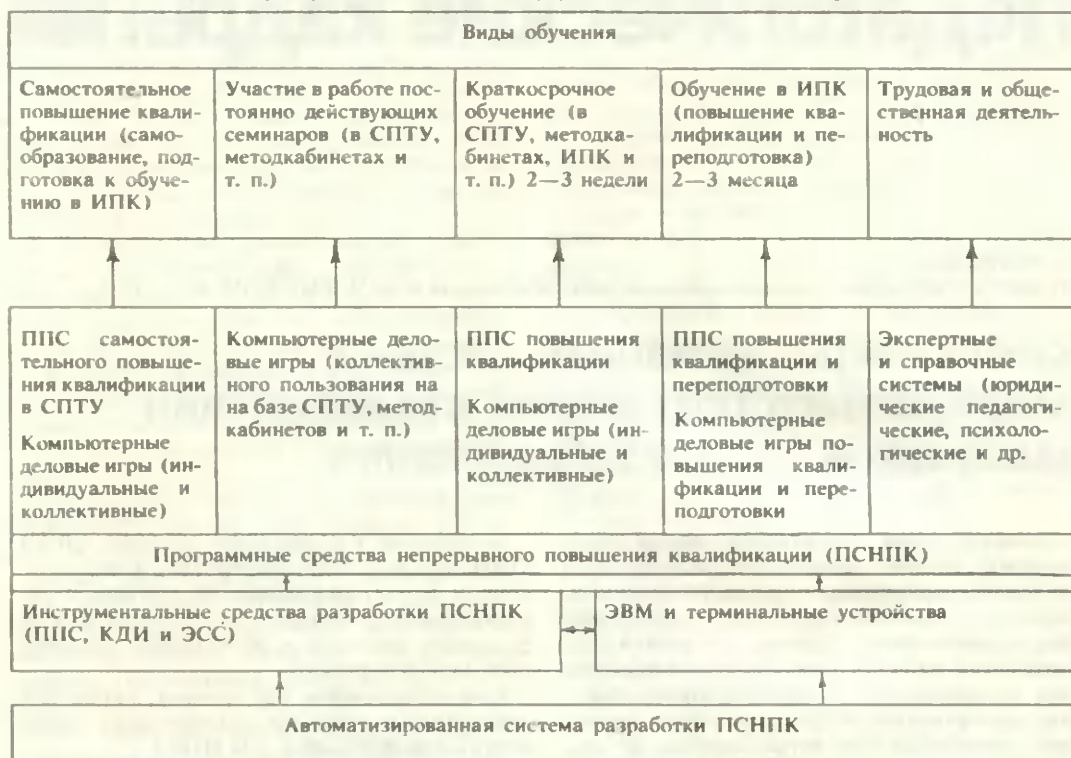
Они объединены под общим названием: программные средства непрерывного повышения квалификации (ПСНПК).

Для разработки прикладных программных средств — сложной и трудоемкой задачи — требуется создать автоматизированную систему, достаточно «дружественную» к преподавателю, разработчику этих средств, и дающую ему возможность (практически без знаний языка программирования) переводить фактические материалы и знания в обучающие программы с помощью специальных инструментальных средств.

Совокупность прикладных ПСНПК, автоматизированной системы их разработки, включающей в себя инструментальные средства разработки ППС, КДИ, ЭСС и технические средства (ЭВМ, терминальные устройства, сети и др.) составляет компьютеризированную систему непрерывного повышения квалификации (КСНПК).

Остановимся на некоторых вопросах, связанных с созданием КСНПК.

В настоящее время разработка ПСНПК проводится централизованно высококвалифицированными преподавателями и работниками ИПК с учетом таких принципов непрерывного образования, как плановость, поступательность, интегративность и преемственность [2]. При этом реализуются компенсаторная (ликвидирующая пробелы в образовании), адаптивная (приводящая знания работника к современному уровню) и, частично, развивающая (обогащающая творче-



ский потенциал человека) функции непрерывного образования [2]. Больше внимание должно быть уделено адаптивной и развивающей функциям. Кроме того решается вопрос управления непрерывным процессом повышения квалификации и реализуется обратная связь, без которой живучесть любой системы весьма низка.

Разработка ПСНПК должна вестись с учетом особенностей познавательной деятельности взрослого человека, где особую роль играют мотивационно-ценностные отношения. Другой особенностью является учет восприятия взрослым человеком информации, в частности с помощью ЭВМ. При этом важными являются такие вопросы обучаемости взрослых, как: «...их способность к продуцированию и репродуцированию новых идей, методов работы, темпа их продвижения в познавательной деятельности, гибкость ума, способности изменять сложившиеся стереотипы, а также вопросы доступности (трудности) содержания для различных категорий людей» [2].

При разработке ПСНПК необходимо использовать методы активизации обучения, характеризующиеся следующими признаками:

вынужденной или побуждаемой активно-

стью, тесно связанной мотивацией;

позапным контролем знаний;

тестированием обучаемого, позволяющим учитывать его индивидуальные характеристики;

наличием прямой и обратной связи обучающегося со средством обучения или преподавателем (через средство обучения).

Экспертные и справочные системы (ЭСС) являются обучающими системами более высокого порядка, чем ППС и деловые игры. Это системы, направленные на практическое решение производственных задач. Учитывая все возрастающий объем знаний, требующийся для решения различного вида практических задач, ЭСС могут стать надежным и эффективным помощником для любого работника системы профтехобразования. Реализация знаний и опыта наиболее квалифицированных специалистов в виде таких систем дает возможность более быстрого их распространения, в первую очередь за счет безбумажной формы передачи информации. Наиболее перспективными являются «открытые» ЭСС, в которых по мере появления новых знаний происходит корректировка базы знаний. Они дадут возможность наиболее опытным и творческим работникам накапливать и распространять свой передовой опыт.



Компьютерные деловые игры являются эффективным средством обучения с высоким уровнем мотивации и ценности знаний, так как фактически проигрывают реальные производственные ситуации при непосредственном участии обучающихся. При этом необходимо отметить, что это сложный и трудоемкий способ, базирующийся на усвоении определенной суммы знаний и, следовательно, являющийся как бы вторым этапом обучения, доводящим теоретические знания до практической реализации.

Работы по созданию компьютерных деловых игр и ЭСС для системы повышения квалификации профтехобразования находятся в настоящее время в начальной стадии, а именно на этапе создания базы знаний, содержащей факты и эвристические правила планирования и решения задач, и требуют проведения больших научных исследований.

ППС для самостоятельного повышения квалификации должны быть понятными и четкими, максимально учитывающими все возможные вопросы и запросы обучаемого, так как эти средства используются без преподавателя. Они должны быть результатом тщательной отработки ППС, используемых в ИПК и включать в себя справочник, словарь и другие вспомогательные средства. Управление ими должно быть предельно упрощено и не вызывать затруднений при самостоятельной работе. Содержание ППС определяется автором курса с учетом принципов поступательности и преемственности непрерывного образования. Основные этапы разработки ППС подробно рассмотрены в [3, 4].

Немаловажным вопросом при разработке ППС является определение их качества. На сегодняшний день методик оценки качества не существует, но основой при их разработке должно быть, в частности, выполнение требований, вытекающих из специфики образования взрослых (ценность знаний, уровень мотивации, доступность знаний). Необходимо также учитывать обоснованность разработки. Для количественной оценки этих показателей можно использовать метод экспертных оценок. Главным критерием каче-

ства являются результаты внедрения ППС в процесс повышения квалификации.

Создание КСНПК профтехобразования описанной выше структуры даст возможность:

реализовать основные принципы и функции непрерывного образования и интегрировать учебную, трудовую и общественную деятельность в единый процесс непрерывного профессионального совершенствования руководящих и инженерно-педагогических кадров; применить в процессе непрерывного повышения квалификации активные и интенсивные методы обучения;

помочь работникам профтехобразования в решении практических задач с помощью ЭСС; автоматизировать разработку ПСНПК; осуществить безбумажную форму распространения научно-педагогических знаний и передового педагогического опыта;

упростить для работников профтехобразования поиск учебных материалов при самостоятельном повышении квалификации и подготовке к повышению квалификации в ИПК;

дать объективные средства контроля уровня знаний и профессиональной квалификации работников профтехобразования в виде контролирующих программ;

эффективно решить задачу освоения вычислительной техники работниками профтехобразования;

частично переместить процесс повышения квалификации в профтехучилище, используя вычислительную технику не только для обучения учащихся, но и для повышения квалификации работников училища.

#### Литература

1. Маркс К., Энгельс Ф. Соч., т. 3. С. 75.
2. Теоретические основы непрерывного образования / Под ред. В. Г. Онушкина. М.: Педагогика, 1987.
3. Марков С. Разработка сценариев педагогических программных средств (методические рекомендации). Л.: ВИПК РР и С ПТО, 1986.
4. Марков С. Разработка сценариев занятий с использованием ЭВМ в системе повышения квалификации (методические рекомендации). Л.: АПН СССР, НИИ ООВ, 1988.

М. СТЕРНИН

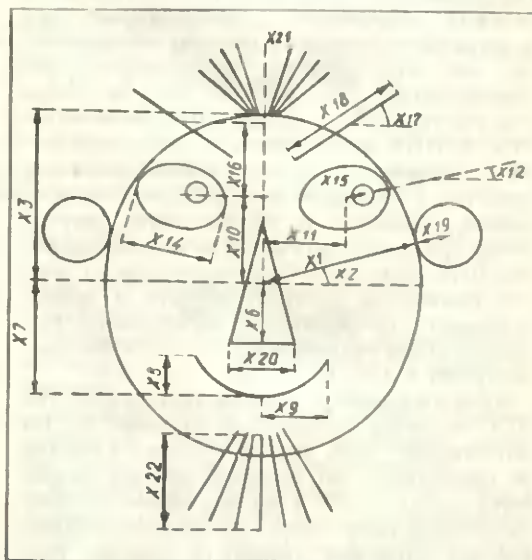
## Тысяча лиц

Правая бровь поползла вверх, зрачки расширились, нос заметно удлинился, опустились уголки рта, появились морщины, и начала расти борода, становясь все гуще и длиннее. «Эк тебя перекосило», — прошептал оператор, не отрывая взгляда от видеоэкрана, на котором гримасничала странная физиономия. И вслед за тем принялся нажимать на клавиши управления, стараясь придать лицу на экране благообразное выражение...

Подобная сценка вполне могла разыгаться там, где ход и состояние сложных процессов отображают графически с помощью так называемой техники лиц Чернова. Появление и растущая популярность этой техники связаны не только с бурным развитием машинной графики, но и с психологическими особенностями нашего восприятия: нам проще понять и оценить многомерную информацию, когда она привычна и наглядна.

С чем вы предпочитаете иметь дело: с изображениями, картинками, графиками или с числовыми таблицами? Почти наверное можно сказать — с изображениями, особенно в тех случаях, когда необходимо дать качественную оценку, сравнить, сгруппировать, классифицировать результаты наблюдений. Наиболее привычны, да и удобны, двумерные графики, нарисованные мелом на доске или карандашом на бумаге. Мы привыкли к ним со школьной скамьи. Есть ли более удобный способ, позволяющий наглядно отразить, как зависит пройденный путь от времени, а скорость протекания реакции — от соотношения масс двух веществ? Однако если число компонентов, участвующих в химическом процессе, велико, то простым графиком уже не обойтись. Возникает непростая

Перепечатывается из журнала «Химия и жизнь».



На схеме — один из способов, помогающих связать характеристики изучаемого процесса (X1—X22) с теми или иными частями физиономии. Например, параметр X1 определяет ширину лица, X17 — наклон бровей, а X22 — длину бороды.

задача — как представить процесс в целом и поведение отдельных его составляющих; говоря более строго, это и есть задача графического отображения многомерной информации.

До чего же сложными получаются при этом графики! Они превращаются в запутанные картины, разобраться в которых, пожалуй, еще сложнее, чем в числовых таблицах. Вот если бы можно было отобразить наглядно и многокомпонентные процессы... В этом нет ничего нереального, методы

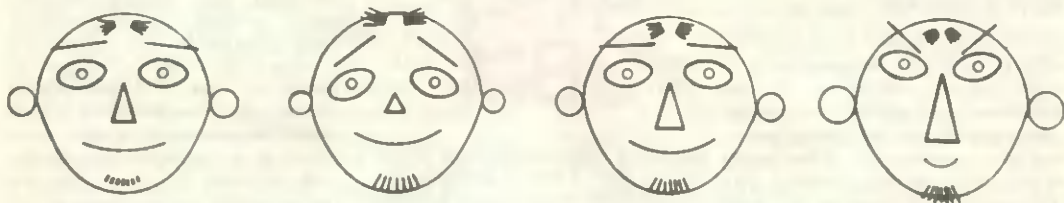
графического изображения многомерных данных существуют; тот, о котором здесь рассказано, пожалуй, самый наглядный и выразительный при всей своей математической строгости.

В 1973 г. американский математик Г. Чернов предложил метод, названный впоследствии его именем, суть которого в том, что комплекс данных отображается в виде стилизованной человеческой физиономии. Каждая черта такого портрета, будь то длина носа или наклон бровей, поставлена в соответствие с характеристиками изучаемого явления. Скажем, чем выше температура реакции, тем сильнее вытягивается физиономия. Или растут уши. Во всяком случае, любое изменение параметров вызывает деформацию лица, придает ему то или иное характерное выражение. Сбой в процессе — и вот вместо наивно улыбающегося подростка мы видим грозно насулпенного старца...

Техника лиц Чернова привлекательна прежде всего потому, что каждый из нас способен мгновенно распознавать малейшие изменения в выражении человеческого лица, даже если это лицо — рисованная карикатура. Конечно, восприятие такого изменения будет сугубо индивидуальным: одному большие глаза покажутся признаком довольства, другому — удивления, третьему — страха, но всякий мгновенно заметит, что собеседник «сделал большие глаза». На протяжении всей нашей жизни мы вглядываемся друг в друга,

начал обрастать бородой и поднимать брови. Оператор мгновенно заметит эти изменения, и не только заметит, но и свяжет их — разумеется, при некотором навыке, после обучения — с теми или иными сдвигами в контролируемом процессе. Зная, что размер носа соответствует давлению в системе, и заметив, что нос принялся расти, оператор первым делом сбросит давление — до того уровня, при котором нос примет разумные размеры; а выяснением причин можно будет заняться позже. Чтобы ускорить и упростить сравнение параметров, эталонное и текущее состояние нередко отображают на правой и левой половинах одного и того же лица: в этом случае всякий перекося легче обнаружить.

Мы не будем обсуждать здесь технические системы и математическое обеспечение, позволяющие реализовать технику лиц Чернова. Заметим только, что необходимо, во-первых, передать информацию об изучаемом объекте к вычислительной машине, во-вторых, придумать и изобразить выразительную физиономию и, в-третьих, поставить в соответствие тем или иным параметрам характерные размеры лица. Последнее требование не так просто, как может показаться: между текущим состоянием процесса и общим выражением лица должна существовать взаимосвязь. Например, сияющая, безмятежная улыбка соответствует нормальному ходу процесса, но выражение становится уг-



#### Сравнение деятельности нескольких предприятий по экономическим показателям.

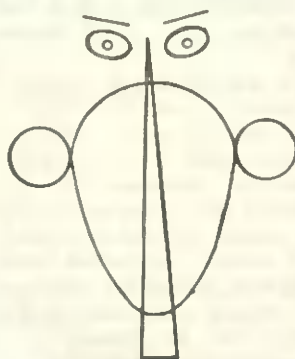
и человеческое лицо стало для нас одним из самых распознаваемых объектов. Именно эти свойства — легкость и надежность сопоставления различных выражений человеческого лица — позволяют быстро решать сложные задачи сравнения многокомпонентных объектов, состояние которых может зависеть, например, от двух десятков параметров.

Представим себе оператора, управляющего технологическим процессом. Перед ним на экране два портрета. Один из них эталонный, он соответствует нормальному или идеальному состоянию процесса, второй отражает текущее состояние. И вдруг второй портрет

рюмым, как только нарушился технологический режим.

Иллюстрации к этим заметкам показывают некоторые возможности техники лиц Чернова; они заимствованы из статьи А. Ф. Роозе и Е. И. Арикеше в журнале «Управляющие системы и машины» (1986. № 3). Отметим, что метод позволяет решать разнообразные задачи классификации и группировки многомерных объектов. Созданный в нашей стране пакет программ ГРАЛ с успехом применялся, по сообщению его авторов, для сравнения качества программных средств ЭВМ, экономических показателей работы промышленных предприятий. Его же использовали, когда требовалось изобразить наглядно, как изменяются во времени свойства объектов.

Требования к точности решения должны быть разумными, а правила конструирования физиономии обязаны отвечать этим требованиям. В противном случае глаза могут не только вылезти на лоб, но и двинуться дальше...



Современная техника позволяет развивать технику «тысячи лиц», вводя цветовую гамму и звук, строя мультипликационные сценарии. Пользуясь специальными средствами интерактивной графики, оператор может при необходимости выбрать для более пристального изучения отдельные части лица, увеличить их до требуемых размеров и оценить, таким образом, тонкие и тончайшие детали многосложных процессов. А если понадобится

все, всегда есть возможность перейти к более привычным видам отображения информации — к таблицам, тексту, речи.

Оператор внес поправки в режим, отдал команду управляющей системе — и вскоре морщины на физиономии разгладились, борода исчезла, рот растянулся в улыбке. С экрана смотрела веселая, чисто выбритая рожица, и оператору показалось, будто она подмигнула — все в порядке.

## Контроль или слежка?

Все большая часть канцелярской работы в США выполняется с помощью компьютеров, и все чаще они используются не только как помощники человека, но и как его контролеры. Компьютер может подсчитать, сколько раз машинистка ударила по его клавишам или сколько покупателей кассир обслужил за смену, сколько времени требуется банковскому служащему на оформление одной операции или как долго клиенту приходится ждать ответа на звонок в справочную службу авиакомпании.

Хозяева фирм полагают, что такой контроль допустим, но профсоюзы возражают против подобных мер, утверждая, что они чреваты вмешательством в личную жизнь работников и приводят к росту вызываемых стрессом заболеваний (неврозы, язвенные болезни, гипертония).

Эта проблема была исследована Управлением по технической оценке при конгрессе США, ситуация признана тревожной.

## ЧТО? МОЖЕТ ЭВМ

Однако в настоящее время планируется ввести запрет лишь на прослушивание телефонных разговоров служащих без специального предварительного сигнала, оповещающего о подключении к линии.

## Об одном фиаско

Мало кто до сих пор считает ЭВМ «электронным мозгом». Специалисты называют ее обычно «дура железная», и не без оснований. Широко известна, например, история о том, как машина требовала оплатить счет на 0 долларов 00 центов. Однако недавно стал известен случай, свидетельствующий о том, что и среди причастных к электронике людей подсознательный пиетет по отношению к компьютеру еще не изжит...

На одном вычислительном

центре (не названном по причинам, которые прояснятся в дальнейшем) был введен «плавающий» график работы: сотрудники должны были отработать за неделю 41 час, а как — все равно. Впрочем, остались ограничения, накладываемые КЗОТом: смена не длиннее 12 ч, если смена более 6 ч — обязателен обеденный перерыв и т. п. Графики составлялись еженедельно, и еженедельно администрации ВЦ приходилось их проверять: а не жульничают ли работнички?!

Однажды администрацию осенило: передовать проверку правильности составления графиков ЭВМ. Составили программу, и вот уже компьютер выдает комментарии: «Иванов — недоработка 1.5 ч», «Петров — среда: выход на 9 ч без обеда». Администрация, жутко довольная, занялась прочими административными делами, полностью положившись на железного цербера.

Итог же этой истории таков: программисты, записываясь в график более чем на 6 часов, в целях экономии времени ставят: «Обед с 12.00 до 12.00». Машина довольна. Люди тоже.

В. ТОЛСТОШЕЕВ

## Автоматизированное рабочее место арбитра

Действующий с января 1988 г. Закон о государственном предприятии (объединении) значительно расширил самостоятельность, заинтересованность и ответственность за рациональное ведение хозяйства основных звеньев экономики. В этих условиях возрастают требования к их связям, в развитии которых (по разным причинам) нередко возникают сбои, что приводит к хозяйственным спорам. Их разрешают органы арбитража.

Определенный объем информационных, технико-юридических, организационных и иных действий при этом с успехом может быть передан ЭВМ. В аппарате Министерства промышленности средств связи СССР функционирует автоматизированное рабочее место арбитра (АРМ), накоплен определенный опыт применения электроники, АРМ оснащено отечественной ЭВМ «Искра-226» со встроенным дисплеем, накопителем на магнитных дисках, диском и печатающим устройством.

АРМ обеспечивает выполнение важных операций, осуществлявшихся ранее непосредственно арбитром. Теперь почти половину трудового дня он проводит за клавиатурой ЭВМ, а остальное время тратит на подготовку и рассмотрение хозяйственных споров: еженедельно от 30 до 40 дел.

Более полное представление о значении и объеме выполняемых АРМ функций дает общее знакомство с характером и масштабами деятельности по разрешению хозяйственных споров. Их рассмотрение ведется на основе заявлений (исков) предприятий и организаций об удовлетворении конкретных требований к контрагенту, начиная заключением договора и кончая возмещением причиненного ущерба. Содержание искового

заявления и процедура его движения определены специальными правилами. Оно должно представляться письменно в арбитраж за подписью руководства предприятия, включать реквизиты свои и другой стороны, причины иска, доказательства, сведения об ответе другой стороны на предъявленную претензию и мотивы несогласия с ним истца.

Все материалы, используемые в дальнейшем при разрешении спора, подлежат регистрации, учету, обработке и хранению. Слушание каждого дела требует подготовительной работы. Арбитрам приходится получать и обрабатывать дополнительные сведения по делу, направлять извещения спорящим сторонам о времени рассмотрения спора, вызывать участников арбитражного процесса. После разрешения дела требуется подготовить и отправить протоколы и приказы нужным адресатам.

Нередко проигравшая сторона подает жалобу после рассмотрения дела, обосновывает свое несогласие с вынесенным арбитром решением, что влечет за собой очередной цикл работ, многие из которых нетворческие, отнимают у арбитров массу времени, отвлекают от решения главной задачи — помощи предприятиям в реализации их законных прав и интересов.

АРМ позволяет вводить в ЭВМ сведения об исковых заявлениях предприятий и организаций, которых только за 1987 г. в аппарат Министерства промышленности средств связи СССР поступило свыше 2,5 тыс. Каждое из них содержит в среднем до 150 байт информации. На регистрацию заявления сейчас уходит не более 1—2 мин, а при ведении традиционного журнала на это тратилось много больше времени. Данные из исковых заявлений вводятся в

ЭВМ почти по 30 позициям, начиная номером заявления и кончая реквизитами отзыва ответчика. В машине фиксируется информация о характере и сумме предъявляемого иска, размерах санкций, дате его направления и регистрации и др.

Одновременно обеспечивается группировка сведений о видах споров: понуждение к заключению договора, просрочка заключения, об изменении условий договора, о расторжении договора, споры о недопоставках, нарушении качества, некомплектности, неправильных расчетах т. д. Дифференцируется информация о причинах возврата исковых заявлений, результатах рассмотрения дела, взысканных суммах, размерах штрафа и особо перечисляемых в союзный бюджет, времени отправления протокола, о датах заявления о пересмотре дела и др.

Достоинство АРМ не только в облегчении труда арбитров и оперативности регистрации исковых документов, но и в полноте собираемых сведений и возможности их дальнейшего использования. В ЭВМ вводится такая информация, которая ранее отсутствовала, в частности сведения о стоимости недопоставленной продукции, сумме штрафных санкций, статистике заявлений о пересмотре принятых арбитром решений, данные об отзывах, грубых нарушениях договорной дисциплины. С помощью ЭВМ готовятся также извещения о назначении дела и сопроводительные письма при возврате неправильно оформленного материала или подготовленного с иными нарушениями. ЭВМ приняла на себя распечатку различных оформительских бумаг (извещений, писем и т. п.), значительно сократив объем машинописных работ. В настоящее время они сохранены лишь для подготовки протоколов рассмотрения дел и приказов о принудительном взыскании сумм, которые в принципе также могут быть переданы электронике. Кстати, бланки этих приказов уже изготавливаются с помощью машины и при согласии работников Госбанка их можно использовать в практической деятельности.

АРМ обеспечивает ввод и оперативный поиск отзывов ответчиков с учетом установленного срока их представления. ЭВМ взяла на себя также функции подборки необходимой информации об участвующих в спорах ответчиках и истцах, обобщения сведений по конкретным предприятиям (за соответствующий период — месяц, квартал и т. п.), участвующим в арбитражном процессе, о решениях по отдельным категориям споров. Ранее эти операции отнимали у арбитров много времени, а некоторые из них совсем не выполнялись из-за трудоёмкости. Они требовали постоянного обращения к ар-

хивным документам, различным материалам, проведения определенных расчетов. К тому же некоторые нужные сведения невозможно было получить, поскольку данные о них вообще не фиксировались в регистрационном журнале.

Автоматизирована подготовка ответов на запросы предприятий о состоянии дел с их участием. Сейчас трудовые коллективы имеют возможность быстрее получать интересующие их сведения о дате рассмотрения дела, о наличии отзыва на иск и, главное, о вынесенном арбитром решении. Ведь дела не всегда слушаются в присутствии представителей трудовых коллективов, приглашаемых обычно в ведомственный арбитраж в исключительных случаях (большая сумма иска, технические вопросы и т. п.). В перспективе, конечно, с помощью телекоммуникаций предприятия смогут участвовать в процессе, не выезжая в центр, где размещается ведомственный арбитраж. Пока же его представитель непосредственно участвует в разрешении хозяйственных споров.

Машина оперативно подбирает информацию по делам, справки о которых направляются в вышестоящие органы, обобщает данные о конкретных видах нарушений за соответствующий период, дает необходимую статистику об арбитражной работе в отрасли.

Содержание и последовательность работ в процессе решения задач на АРМ устанавливаются специальным руководством пользователя. Комплекс программ обеспечивающих работу АРМ, охватывает следующие функции: регистрацию поступающих в арбитраж дел, корректировку и дозапись реквизитов регистрируемых документов, формирование и вывод на АЦПУ карточек учета дел по участникам спора (истец, ответчик).

Решение задач не требует участия оператора. Он обеспечивает лишь ведение массивов условно-постоянной информации (классификаторов), описание форм ввода-вывода, дублирование массивов базы данных. Комплекс программ реализует диалоговый режим пользователя с выбором конкретной функции путем последовательного формирования запросов на выполнение соответствующей программы. Контроль правильности кодовых обозначений осуществляется на экране.

Время обучения пользователя (неспециалиста по программированию) не превышает двух недель. Операции выбираются самим пользователем из приведенного перечня на экране. После включения ЭВМ и подготовительных операций он переходит с помощью клавиатуры к выбору из базы данных.

Применение компьютеров в арбитражном

процессе предъявляет повышенные требования к его организации и документальному оформлению. Пока материалы, присылаемые предприятиями и объединениями, не всегда точны и полны, а от этого снижается эффективность автоматизации. Отсутствие, например, в отзыве ответчика номера искового заявления исключает автоматическое присоединение к последнему поступающего отзыва, приводит к лишним затратам машинного времени на поиск номера в памяти ЭВМ. Если выборка нужной информации при надлежаще оформленной документации занимает десятки секунд, то при отступлениях приходится тратить времени в десять раз больше, поскольку требуется перебирать дополнительные позиции.

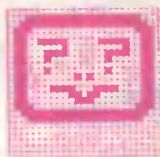
Автоматизация арбитражной работы, призванная улучшать качество и повышать оперативность рассмотрения хозяйственных споров, не снимает проблему преодоления формального подхода к реализации предоставленных предприятиям широких полномочий. Еще встречаются ситуации, когда на рассмотрение хозяйственных споров требуется значительно больше средств, чем сама сумма спора. Руководители отдельных предприятий порой не желают урегулировать возникающие конфликты, особенно на стадии заключения договоров, затягивают направление отзывов на поступившие исковые заявления. Вместо установленного максимального пятидневного срока представление отзыва оттягивают иногда до месяца и более. Прав арбитража по применению к недисциплинированным предприятиям штрафных

санкций за неправильное оформление отзывов, непредставление запрашиваемых дополнительных материалов становится явно недостаточно.

Значителен еще процент в общей массе дел так называемых бесспорных исков. Порой перед самым рассмотрением дела ответчик сообщает о признании иска или добровольном перечислении соответствующих сумм на счет истца, согласии на принудительное списание со своего счета конкретной суммы. Несмотря на то что бесспорные иски отнимают много времени у ведомственного арбитража, нет действенных рычагов воздействия на сокращение их потока.

Повышению отдачи ЭВМ могла бы способствовать оплата предприятиями затрат, связанных с рассмотрением в аппарате министерства исков, чтобы они не обращались с делами на мизерные суммы, несли материальную ответственность за бесспорные иски. Как известно, определенная пошлина в зависимости от суммы иска взимается органами вневедомственного (государственного) арбитража, но она не установлена для разрешения хозяйственных споров в отраслях. Видимо, установление такой пошлины при рассмотрении хозяйственных споров ведомственным арбитражем отвечало бы современным принципам хозяйствования. Перечисляемые предприятиями суммы частично могли бы вноситься в бюджет, а также частично расходоваться на совершенствование юридической работы в народном хозяйстве, в том числе на ее компьютеризацию.

**Веселый урок**



**Письмо в редакцию**

*Приходится констатировать, что наш отечественный словарь информатики и вычислительной техники сильно замусорен словами иностранного происхождения, чисто техническими или узкоцеховыми терминами, а то и просто жаргоном.*

*За примерами далеко ходить не надо: программирование, информатика, компьютер, ассемблер, алгоритм, дамп, опция, интерфейс, драйвер — и так далее и тому подобное.*

*Нами разработан оригинальный понятийный аппарат, который нагляден, образен и легко доступен для понимания. Пред-*

*лагаем его вниманию читателей журнала «Информатика и образование».*

*С уважением  
д-р И. В. СИДИСИДЗЕ,  
г. Дубна*

Считало  
Запоминало  
Ругало  
Пробивало  
Жевало

Казало  
Крутило-запоминало  
Ходило-приставало  
Ходило-рассыпало  
Моргало  
Нажимало  
Рисовало  
Зависало

Процессор  
ОЗУ  
Транслятор  
Перфоратор  
Устройство ввода перфокарт  
Дисплей  
Диск  
Пользователь  
Оператор  
Курсор  
Клавиатура  
Графопостроитель  
ОС БС

## «Мертвая зона», или Павильон № 2

С февраля по май этого года на ВДНХ СССР проходила выставка «Технические средства обучения, приборы и оборудование для высшей и средней школы».

Побывать на выставке пришлось трижды. Интересовало меня прежде всего то, что относится к информатике. Выставка освещалась и в прессе, и по телевидению, поэтому ограничусь лишь несколькими словами об одном ее аспекте. Его отмечают многие, побывавшие на ней.

Раздел дошкольного воспитания. Обращает на себя внимание компьютерно-игровой комплекс для малышей. Изготовитель этого, пока единственного образца, — ЦНИИ «Электроника». Пакеты программ, разрабатываемые в НИИ дошкольного воспитания АПН СССР «призваны формировать у детей первоначальные навыки работы с ЭВМ». Но ни детский смех, ни взрослая речь не нарушали тишину около расположившегося среди игрушек на столике-подставке экспоната. Побеседовать о перспективах компьютерного всеобуча дошкольников не с кем. В следующее посещение столик был пуст, экспонат исчез. Хорошо хоть фотография осталась! (фото 1.)

Воплощение творческих усилий разработчиков, надежда преподавателей информатики — дисплейные классы «Корвет» и «УКНЦ» притягивают глаз яркой неэргономичной окраской. Но согласитесь, когда школьный учитель, живущий за тысячу верст, едет в столицу посмотреть разрекламированные дисплейные классы, а видит безмолвные ряды «железок», и не где-нибудь, а на ВДНХ СССР, это никуда не годится! Что он сможет рассказать о перспективах технического оснащения компьютерного всеобуча ожидающим его коллегам и ученикам? На худой конец он мог бы привезти проспекты, но нет и их. Конечно, работа этих классов наверняка демонстрировалась, и не раз, но кому? За все три посещения (одно из них в дни весенних каникул) людей на территории «мертвой зоны» я так и не увидела.



1



2

Может быть, программное обеспечение и методические рекомендации по его использованию — то, что действительно следовало демонстрировать на выставке, — важная государственная тайна? Не хочется думать, что этого еще просто нет.



3

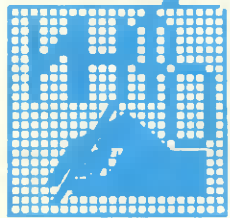
Хранили программное безмолвие и другие электронно-вычислительные экспонаты: разнообразные промышленные и несерийные ПЭВМ. Робототехнические комплексы — новое и интересное направление использования ЭВТ в учебном процессе — не были исключением (фото 2).

Запертый автобус с зашторенными окнами (фото 3). Интересно, «для кого» на выставке был закрыт передвижной класс вычислительной техники? Возможно, внешний (или внутренний) вид компьютеров, установленных вместо привычных сидений, — коммерческая тайна?

Напрашивается вопрос: не целесообразнее ли сократить сроки проведения выставки, но зато обеспечить обязательную демонстрацию работы экспонатов для всех заинтересованных посетителей в часы ее работы? Словом, лучше меньше, да лучше. А вот проспектов пусть будет больше.

А. КРАВЦОВА





## Срочно требуется задачник!

*Лучший способ узнать —  
это сделать.*

Г. М. Клейман

Обучение информатике, как и любому школьному предмету, неразрывно связано с выполнением различных упражнений и решением задач. Попробуем определить, какими должны быть эти упражнения и задачи, в какой последовательности и в каком количестве предлагаться учащимся, каков должен быть уровень сложности таких заданий.

Несмотря на многочисленные споры, предмет и методы обучения информатике, по-видимому, неразрывны с предметом и методами обучения программированию. В вышедшей недавно книге «Школы будущего: компьютеры в процессе обучения» [1] автор пишет: «...большинство людей, использующих компьютеры, не пишут своих собственных программ, и им практически вообще не требуется знать программирование. Однако изучать программирование полезно. Изучая программирование, ученики лучше понимают сущность работы компьютеров, их возможности и ограничения. Программирование помогает школьникам развивать навыки мышления и решения задач, а также привычку к аккуратной работе. Кроме того, многим ученикам нравится управлять работой компьютера при помощи своих собственных программ».

Итак, будем исходить из того, что программированию учить надо. Но как?

В многочисленных пособиях по программированию на основе конкретных языков реализуются в основном два подхода к обучению, которые так характеризует Б. Шнейдерман [2]: «При классическом подходе внимание сосредоточивается на выработке синтаксических знаний и на обучении кодированию, в то время как при подходе, основанном на решении задач, дают семантические (курсив мой. — Е. К.) знания и учат решать задачи на высоком уровне... Ни один из этих подходов не является правильным, при их использовании просто преследуются различные цели».

Однако это две стороны одной медали и друг от друга они неотделимы. Недаром многие специалисты (Н. Вирт [3], Р. Форсайт [4] и др.) отмечают большое влияние выбранного языка программирования на формирование и стиль мышления, а Д. Грис призывает [5]: «Программируйте для языка программирования, а не на нем».

По всей видимости, изучение идей программирования должно проходить параллельно с изучением языка для выражения этих идей, причем следует идти от задачи к правилам языка, а не наоборот.

При этом синтаксис языка необходимо закреплять упражнениями типа: найти синтаксическую ошибку в записи, выбрать из предложенных записей синтаксически правильную, дополнить конструкцию, определить, что напечатает программа и т. п. Однако обучение синтаксису — самая простая часть работы учителя. Значительно сложнее передать учащимся идейное содержание предмета.

Ч. Узерелл пишет [6]: «Преподавание программирования — дело почти безнадежное, а его изучение — непосильный труд... все усилия тщетны, если студент не будет практиковаться в написании программ». То есть при обучении программированию решение задач имеет совершенно особое значение, потому что программированию можно научиться только на практике.

Роль учителя в этой работе можно сформулировать словами Г. М. Клеймана [1]: «Необходимо способствовать тому, чтобы ученики проводили систематический анализ задач, решаемых их программами; следовали принципам структурного программирования; уделяли должное внимание деталям написания программ, их тестированию и отладке».

Задачник по программированию должен в первую очередь помочь сформировать и закрепить семантические знания учащихся.

На начальной ступени обучения он предназначен для овладения совокупностью стандартных приемов и методов, которые необходимы как инструмент для последующей творческой работы.

Кроме того, нужно, чтобы учащиеся осознали

ЭВМ как универсальный исполнитель. Для этой цели очень полезны задачи программирования различных исполнителей с фиксированными системами команд.

Невозможно отделить от обучения программированию обучение использованию языков программирования средств, например, текстовых редакторов, баз данных, электронных таблиц.

Все эти потребности курса должны быть поддержаны развитой системой задач. Хорошо написанный задачник по программированию для школы на сегодняшний день может сыграть роль, вполне сравнимую с учебником информатики.

## Литература

1. Клейман Г. М. Школы будущего: компьютеры в процессе обучения. М.: Радио и связь, 1987.
2. Шнейдерман Б. Психология программирования. М.: Радио и связь, 1984.
3. Вирт Н. Алгоритмы + структуры данных = программы. М.: Мир, 1985.
4. Форсайт Р. Паскаль для всех. М.: Мир, 1987.
5. Грис Д. Наука программирования. М.: Мир, 1984.
6. Уззерелл Ч. Этюды для программистов. М.: Мир, 1982.

Е. КРЫЛОВА

## ”Имитация с целью симуляции”

120

Уважаемая редакция! В вашем журнале (№ 2 за 1986 г. и № 1 за 1987 г.) были помещены статьи «Какой должна быть обучающая программа?» и «Практикум по курсу ОИВТ». Обе задуманы как ключевые, основополагающие работы, и обе, к сожалению, нуждаются в критическом разборе.

«Какой должна быть обучающая программа?»

Эта статья весьма трудна для восприятия, содержит массу несурзностей и витиеватые выражения наподобие «имитационно-моделирующая программа, предназначенная для «симуляции» процессов, явлений». Рассмотрение ее в целом выходит за рамки письма, поэтому остановлюсь на анализе отдельных утверждений.

Статья посвящена педагогическим программным средствам (ППС). Определение этому термину не дано, однако отмечено, что «ППС является качественно новым средством обучения, обладающим уникальными возможностями...», в частности позволяющим достичь следующих методических целей:

1) «осуществление контроля с обратной связью, с диагностической и оценкой результатов (или без них)»;

2) «моделирование и имитация изучаемых или исследуемых процессов, явлений с переходом реальности—модель и наоборот (или без перехода)»;

3) «проведение лабораторных работ (по физике, химии) в режиме интерфейса с помощью комплекта оборудования, представленного в компьютерной программе»;

4) «обеспечение доступа к сети информации».

Эти возможности не столько уникальны, сколько невероятны. Не нужно быть специалистом, чтобы понимать, что:

1) невозможно осуществлять контроль без диагностики и оценки результатов;

2) невозможно моделировать процессы без перехода реальность — модель;

3) невозможно подключить компьютер к оборудованию, изображенному на экране;

4) сеть информации — такой же нонсенс, как сеть энергии. Если считать, что информационная сеть и сеть информации — это одно и то же, то чем тогда отличается правильное доказа-

тельство от доказательства правильности, а программированное обучение от обучения программированию?

В качестве еще одного примера, характеризующего уровень статьи, отметим утверждение о том, что возможность переноса программы с ЭВМ одного типа на ЭВМ другого типа обуславливается способностью программы «без труда передаваться по локальной сети».

«Практикум по курсу ОИВТ». Эту статью анализировать гораздо проще: она имеет практическую направленность, и достаточно только попытаться принять ее как руководство к действию, чтобы обнаружить — по прямому назначению использовать ее нельзя.

Даже без экспериментальной проверки данного «практикума», исходя лишь из его текста, можно заметить следующее:

на занятиях 1—2 учащиеся не смогут самостоятельно работать с ЭВМ, как это предусмотрено планом занятия (знакомство с клавиатурой ЭВМ должно произойти лишь на следующем занятии);

на занятии 3 учащиеся должны учиться пользоваться клавишами управления курсором, не зная, что такое курсор (понятие курсора вводится позднее);

на занятиях 4—5 учащиеся не узнают, как классифицировать функции (понятие функции вводится позже);

на занятиях 7—9 учащиеся должны изучать понятия текстовой константы и текстовой переменной, еще не имея понятия о текстовой информации, которая вводится на занятиях 14—15;

на занятиях 18—20 учащиеся не смогут решить задачу поиска нечетных чисел, предлагаемую в практикуме (данная задача не может быть решена без использования условного оператора, который вводится позже);

на занятиях 26—29 вводится условный оператор IF — THEN. Вводится вопреки традиционному подходу и логике не перед оператором цикла, а значительно позже. При такой последовательности подачи материала учитель лишается возможности использовать оператор IF — THEN при объяснении оператора цикла.

Перечисленные ошибки довольно просто обна-

ружить и исправить. Гораздо сложнее для учителя корректировать практикум в тех случаях, когда нужно не переставлять, а вычеркивать. Необходимы не только профессиональные знания, но и определенная смелость, чтобы вопреки методической статье продолжать считать, что: клавиши РУС, ЛАТ и РЕГ не являются управляющими, а клавиши УПР и СБР являются таковыми (занятие 3);

Бейсик имеет только 3 режима работы, а не 6; оператор IF — THEN является условным оператором, а не оператором условного перехода.

А что подразумевается под динамикой движения объектов (занятие 23)? Но довольно. Перечисленного достаточно, чтобы вынести этому практикуму должную оценку.

Ущерб, приносимый обществу слабой публикацией, может быть очень велик. Вот пример из методического руководства к таблицам «Алгоритмы и алгоритмический язык», «Алгоритмы работы с величинами». На с. 4 приводится алгоритм прохождения лабиринта с одним входом и одним выходом. Учитель должен объяснить учащимся, что данный алгоритм не приводит к цели, если лабиринт имеет кольцевые коридоры, и предло-

жить самостоятельно построить алгоритм, свободный от этого недостатка. Среднему учителю потребуется не менее 10 мин на то, чтобы разобраться в алгоритме и убедиться в том, что он приводит к выходу из любого лабиринта с одним выходом и одним входом и, следовательно, никакой другой алгоритм не нужен.

Так вот, если умножить напрасно потраченное время (10 мин) на тираж данного пособия (60 тыс. экз.) и на ставку учителя (для простоты 1 руб. в час), то нанесенный ущерб можно оценить в 10 тыс. рублей. И это цена лишь одного неверного абзаца.

Я прошу опубликовать мое письмо, чтобы неискушенный читатель не испытывал почтительного оцепенения перед словом «симуляция». Чтобы те, кто так и не смог понять, как можно моделировать процессы без перехода «реальность—модель», не связывали это с собственной бестолковостью. Чтобы те, кто не знаком с 6 режимами Бейсика, не считали себя профессионально непригодными. У Бейсика только 3 режима. Пусть все это знают.

С. КОРШУНОВ

121

## Из опыта эксплуатации комплексов учебной вычислительной техники

Сейчас в учебных заведениях установлено множество различных ЭВМ — КУВТ «Ямаха», «Агат», «Корвет», КУВТ-86 и т. д. Объединяет их лишь способность доставлять своим владельцам неприятности — крупные и мелкие.

Многие завидуют обладателям КУВТ «Ямаха», не зная, что при заключении контракта на их поставку Минпрос СССР не потребовал включения в комплект принципиальных электрических схем аппаратных средств. В результате борьба даже с незначительными отказами оборудования, которые можно устранить, используя отечественную элементную базу, силами самих учебных заведений, сопряжена с большими трудностями. Увы, большинство специализированных организаций, созданных для ремонта вычислительной техники, всячески отнекиваются от обслуживания импортных машин.

В этих КУВТ слабыми местами являются НГМД, принтер, учительский монитор. Их отказ способен парализовать работу всего КУВТ или же отдельных программ пользователя. О несоответствии ученического монитора санитарным нормам, нехватке дискет и т. д. журнал уже писал на своих страницах. Не раз высказывалось также мнение о нежелательности использования Бейсика в учебном процессе. В программном обеспечении КУВТ «Ямаха» есть компиляторы с Паскаля и Фортрана-77, однако компиляторов недостаточно для эффективной организации обучения этим языкам: необходимо иметь также руководство программиста по данным версиям языков и возможность использования их не только на ЭВМ

учителя. Каждая ПЭВМ КУВТ «Ямаха» оснащена легкосъемной кассетой ПЗУ с интерпретатором Бейсика; видимо, можно было бы заказать такие кассеты с «защитами» в них Паскалем и Фортраном.

Отсутствие тестовых программ для контроля правильности функционирования аппаратных средств данного КУВТ также весьма прискорбно.

Организации, получившие КУВТ-86, столкнутся с неприятностями, частично уже упомянутыми в ИНФО.

Аппаратная реализация и программное обеспечение комплекса далеки от совершенства. Вряд ли оправдано соединение в одном комплексе столь разнотипных ЭВМ, как ДВК-2М и «Электроника БК-0010Ш», имеющих различную архитектуру и программное обеспечение. В КУВТ-86 их объединяет, пожалуй, только набор неприятных черт: низкая надежность электронных узлов, нередко некачественные сборка и монтаж, издаваемый вентиляторами ДВК шум и свист строчной развертки телевизоров, подключенных к БК-0010Ш.

В программных средствах КУВТ-86 следует отметить некорректный компилятор Бейсика и вопиющую неполноту документации по ОС РАФОС.

Совместимость БК-0010Ш и ДВК-2М на уровне команд микропроцессора, конечно, увеличивает возможности КУВТ-86 по сравнению с двенадцатью БК-0010, однако воспользоваться этим рядовой пользователь не в состоянии.

А. ВАСИЛЬЧЕНКО

г. Шуя

## Переподготовка преподавателей по информатике и вычислительной технике

В межотраслевом институте повышения квалификации кадров по новым направлениям развития техники и технологии (МИПК), созданном при Ереванском политехническом институте им. К. Маркса, организованы шестимесячные курсы переподготовки учителей школ, преподавателей СПТУ и техникумов по информатике и вычислительной технике.

Занятия на курсах проводятся с отрывом от производства в течение двух семестров. Первый начинается 1 октября и заканчивается 25 декабря. Затем следует экзаменационная сессия, в течение которой слушатели сдают 4 экзамена. Второй семинар длится с 15 января по

15 марта. Экзаменационная сессия включает 4 экзамена и завершается в марте.

Слушатели, успешно сдавшие обе экзаменационные сессии, допускаются к дипломному проектированию под руководством опытных специалистов. Во время дипломного проектирования слушатели не освобождаются от работы. Защита дипломных проектов проводится во второй половине мая на заседаниях государственной экзаменационной комиссии. При успешной защите слушатели получают дипломы установленного образца.

Особое внимание на курсах уделяется вопросам приобретения практических навыков по ис-

Учебный план

Наименование дисциплин	Всего часов	Лекции	Лаб. зан.	Прак. зан.	Форма контроля	Семестр
Общественно-политический раздел	24	16	—	8	зач.	1
Методы вычислений	72	24	24	24	зач. экз.	1
Технические средства микро-ЭВМ	60	48	—	12	зач. экз.	1
Современные технические средства обучения	36	24	12	—	экз.	1
Проблемно-ориентированные языки программирования	280	128	88	64	курс. пр. зач. экз.	1, 2
Программирование на макроассемблере	64	32	16	16	зач. экз.	2
Операционные системы	48	24	24	—	экз.	2
Автоматизированные обучающие системы	48	24	16	8	курс. пр. зач. экз.	2
Микропроцессорные средства ВТ	32	16	16	—	зач.	2
Основы САПР, ГАП, и РТК	16	16	—	—	зач.	2
Актуальные проблемы психологии обучения	16	16	—	—	зач.	2
Сети ЭВМ	16	16	—	—	зач.	2
Итого	712	384	196	132		

пользованию микропроцессорной техники и работе с ЭВМ. Для этих целей парк ЭВТ распределен таким образом, чтобы каждый обучаемый был обеспечен ПЭВМ на всех практических занятиях.

В институте используются дисплейные классы, оборудованные микро-ЭВМ ЕС-1840, КУВТ «Корвет», ДВК-1, КУВТ-86, ДВК-2М, ДВК-3М, КУВТ «Ямаха» (базовый класс), микро-ЭВМ «Электроника-85», «Искра-1256», «Искра-1030», «Искра-226»; лаборатория микротренажеров К1-20, лаборатория периферийных устройств микро-ЭВМ, лаборатория САПР «Кулон-1» на базе «Электроники-100/25».

Каждая лаборатория имеет от 10 до 16 рабочих мест, что позволяет проводить эффективные практические занятия с подгруппами по 10—12 человек.

Слушатели обеспечиваются благоустроенным общежитием с оплатой в порядке, установленном для аспирантов. На время обучения за слушателями сохраняется заработная плата в размере должностного оклада по месту работы. Иногородним слушателям, получающим заработную плату ниже 140 руб. в месяц, направляющей организацией выплачивается стипендия из расчета, чтобы стипендия и заработная плата в сумме не превышали 140 руб. в месяц.

Расходы по проезду слушателей оплачиваются организацией, направившей их на обучение.

Для организации досуга в институте действует

клуб «Алгоритм». Он регулярно организует общественно-политические и культурно-массовые мероприятия, встречи с артистами театра и кино, диспуты, экскурсии, выставки картин художников и т. д.

Если Вас устраивает предложенная программа, Вы можете поступить на курсы. Для этого необходимо согласовать вопрос с непосредственным руководством, а затем обратиться в соответствующее министерство или ведомство для оформления направления на учебу в МИПК. После получения соответствующей заявки от министерства, МИПК заключает с ним финансовый договор, согласно которому на расчетный счет МИПК за каждого обучаемого перечисляется 660 руб.

В МИПК необходимо отправить следующие документы:

1. Заявление на имя директора.
2. Направление министерства или ведомства.
3. Копию диплома о высшем образовании.
4. Выписку из трудовой книжки.
5. 3 фотокарточки 3×4.

Оформление документов производится с 20 августа по 20 сентября.

При необходимости программы всех курсов будут высланы по соответствующим заявкам.

Наш адрес: 375040, Ереван-40, ул. Ачаряна, 31. Межотраслевой институт повышения квалификации кадров по новым направлениям развития техники и технологии.

123

**И. РОБЕРТ,**

канд. пед. наук, НИИ ШОТСО АПН СССР, ученый секретарь межведомственной комиссии по ППС

## Программно-методическое обеспечение школьных ЭВМ

Преподавание основ информатики и вычислительной техники проводится во всех школах страны, как оснащенных вычислительной техникой, так и не имеющих ее. В связи с этим в настоящее время широко практикуется использование межшкольных кабинетов ВТ, организуемых на базе УПК или общеобразовательной школы. Основной формой организации работы в таких кабинетах являются практические занятия (практикумы). Их цель — формирование у учащихся знаний о возможностях ЭВМ, ознакомление с основными приемами и методами составления простейших программ, выработка умения пользоваться программным обеспечением.

В регионах, слабо оснащенных техникой, практикумы можно организовать на ВЦ. При этом рекомендуется составлять скользящий график проведения практикумов с учетом «пропускной способности» ВЦ таким образом, чтобы на каждую школу приходилось как минимум одна неделя. Такой опыт уже существует в Свердловской и Новосибирской областях.

Как обстоит дело с программно-методическим обеспечением практикумов?

На сегодняшний день сформирован определенный фонд программ, предназначенных для школы: их называют педагогическими программными средствами (ППС). Разрабатываются они учителями, методистами, научными коллективами или отдельными специалистами в НИИ АПН СССР, в институтах АН СССР, в вузах страны.

Сейчас число программ, пригодных для использования в процессе обучения, составляет около 400 для ЭВМ «Ямаха», более 50 — для ПЭВМ «Агат» и 120 — для КУВТ-86.

Каждый практикум содержит программное обеспечение, состоящее из 35—50 программ разнообразного профиля. Среди них программы, обучающие работе на ЭВМ, представляющие устройство ЭВМ и ее узлов, обучающие определенному языку программирования; имеются программы, предназначенные для преподавания учебных предметов общеобразовательного цикла, демонстрационно-игровые программы; инструментальные программные средства для учителя, операционные системы, электронные таблицы, информационно-поисковые системы. Комплект программ в практикуме составляется в соответ-

ствии с его назначением, количеством часов, выделяемых на проведение занятий и типом ЭВМ и оснащается методическими руководствами для слушателей и преподавателя.

В настоящее время при проведении практикумов используются программный комплект «И-86» для ПЭВМ «Ямаха», включающий 37 программ; программно-методический комплекс (ПМК) «Информатика-87» (для ПЭВМ «Агат», КУВТ-86 и «Ямаха»), предназначенный для обеспечения начальной подготовки по ОИВТ учителей и руководящих педагогических кадров системы народного образования. ПМК осуществляет программную поддержку 30 часов практической работы слушателей на ЭВМ и содержит два методических пособия. В одном приведены справочные сведения по используемым программным системам и описание сценариев выполнения всех самостоятельных работ; второе, детально описывающее процесс проведения занятий, содержит лекционный материал по 6 основным направлениям использования ЭВМ в современном обществе.

Летом 1987 г. были проведены Всесоюзные семинары для представителей союзных республик, на которых тиражировались материалы программно-методических комплексов для КУВТ-86 (в Новосибирске и Москве), ЭВМ «Ямаха» (в Новосибирске) и ПЭВМ «Агат» (в Минске). Справку об этих материалах учителя информатики могут получить в своих РИУУ. Материалы ПМК «Информатика-87» и программного комплекта «И-86» были переданы всем пользователям базового КУВТ «Ямаха».

Содержание и состав программного обеспечения практикумов неоднократно обсуждались на Всесоюзном семинаре «Компьютер и образование», который работает в Москве, в помещении президиума АПН СССР в последний четверг каждого месяца. Посещать семинар может любой интересующийся проблемами компьютеризации обучения. На семинаре проводится апробация научной и методической продукции, разрабатываемой по комплексной программе АПН СССР «ЭВМ в школе». В свое время семинар одобрил

практикум как наиболее удобную форму организации практических занятий в школе в условиях ограниченного доступа школьников к ЭВМ.

Для повышения эффективности разработки и качества ППС создана межведомственная комиссия по ППС для школьных ЭВМ, в состав которой входят специалисты различного профиля: учителя информатики, методисты, программисты, гигиенисты и ученые, сфера деятельности которых связана со школьной информатикой. Комиссией определена первоочередная номенклатура ППС: техническая и нормативная документация, регламентирующая порядок организации работ по созданию ППС, приемку ППС. Комиссия рассматривает и инициативные разработки — компьютерные программы с методическими материалами по их использованию. Желающие сдать свои разработки могут обращаться по адресу: 119117, Москва, Погодинская ул., 8, НИИ ШОТСО АПН СССР, ученому секретарю комиссии по ППС, тел. 214-88-35.

В настоящее время разработан первый вариант теоретической концепции программно-методического обеспечения учебного процесса на перспективу: ее первая редакция уже обсуждена в АПН СССР на координационном совещании в апреле и августе 1987 г. На основании теоретических положений концепции для экспериментальной проверки подготовлено свыше 100 программ по различным предметам, из них 40 может быть предложено для промышленного тиражирования в качестве самостоятельных объектов.

С этими программами можно ознакомиться в рабочем порядке на семинаре, о котором упоминалось выше, а также при посещении методистами РИУУ лабораторий НИИ СиМО, НИИ ИВТ, НИИ ШОТСО АПН СССР. Сотрудники лабораторий информатики и вычислительной техники этих институтов регулярно проводят консультации для учителей и методистов по вопросам программно-методического обеспечения практикумов и оборудования кабинетов вычислительной техники. Консультации сопровождаются не только демонстрацией программного обеспечения, но и тиражированием принятых комиссией программ.

## Всесоюзный семинар-практикум

В марте этого года на ВДНХ СССР проходил Всесоюзный семинар-практикум «Вопросы разработки ППС в опорных училищах». Цель его проведения — выяснить состояние дел по разработке и внедрению ППС в опорных профтехучилищах, в которые, согласно разработанному положению, была направлена вычислительная техника. В семинаре приняли участие представители опорных профтехучилищ и методических кабинетов союзных республик, ВНИИ ПТО, ВМЦ, ВИПК, СИПИ и институтов АН СССР. В рамках семинара была организована демонстрация ППС, обсуждены

вопросы, связанные с совершенствованием их качества.

Приведем оценку двух участников семинара, точки зрения которых выражают общие мнения многих выступавших.

С. А. Марков, доцент кафедры ЭВТ ВИПК: «Семинар выявил возросший уровень разработки ППС, но при этом наблюдается снижение творческой активности — прошел, вероятно, первый энтузиазм, и появился трезвый взгляд на ЭВМ не как на панацею от всех бед, а как на средство автоматизации интеллектуального труда. Семинар



позволил каждому из участников оценить свое положение относительно других и определить свои слабые и сильные стороны, обменяться опытом и наиболее интересными ППС. Расширились непосредственные контакты, завязались новые творческие и деловые отношения. Семинар отчетливо выявил общие трудности и проблемы, возникающие у разработчиков ППС и преподавателей, внедряющих ЭВМ в учебный процесс: отсутствие методических рекомендаций по разработке, использованию и оценке качества ППС для СПТУ; отсутствие однозначного взгляда на оплату преподавателей, разрабатывающих ППС, на штаты для училищ, оснащенных ВТ».

Более категорично высказался представитель Казахской ССР, много лет связанный с ВТ, С. К. Сазанбаев: «Рассмотрим задачу разработки ППС не отдельного урока, а всей системы или

курса. По силам ли она опорному СПТУ? И это в условиях, когда нет необходимых инструментальных средств, а для использования имеющегося программного обеспечения нет достаточного количества специалистов высокой квалификации, нет соответствующих средств ВТ. Представляется, что необходимо сосредоточить массовую разработку ППС в отдельных научно-производственных коллективах — дидактических центрах. При этом необходимо предусмотреть широкое участие преподавателей СПТУ в разработке сценариев для ППС с последующей тщательной экспертизой. Тогда прогрессивные формы изложения учебного материала, опыт педагогов-новаторов будут доступны для обобщения и тиражирования в виде ППС. Настало время разработать единую государственную концепцию компьютеризации народного образования. Должна быть выработана единая стратегия внедрения ВТ в учебный процесс, определены организационные структуры по отработке технологии использования компьютера в решении задач интенсификации и индивидуализации обучения.

Всем нам еще предстоит психологическая перестройка в понимании роли и места компьютера в учебном процессе».

Но вот что показалось странным: когда участникам семинара было предложено опубликовать на страницах нашего журнала разработанные ими и представленные на семинар программные педагогические средства (которые они используют и постоянно обмениваются на, так сказать, неформальном уровне), ни один из участников такого желания не высказал. Почему, как вы думаете?

125

## МНР: компьютеризация образования

4—9 апреля в Улан-Баторе состоялось шестое рабочее совещание представителей организаций-исполнителей по проблеме 1.2.7. «Совершенствование системы образования на основе применения средств вычислительной техники (профессионально-техническое образование)» Комплексной программы научно-технического прогресса стран — членов СЭВ до 2000 г. Монголия не случайно была избрана местом проведения такого совещания.

В конце 1986 г. были приняты тезисы МНРП «О совершенствовании народного образования». В них большое внимание уделено вопросам компьютеризации. Во главу угла был поставлен кадровый вопрос. Центром подготовки специалистов стал политехнический институт. С 1987 г. в нем началось обучение по новой специальности «АСУ и вычислительная техника». Сейчас на первом потоке обучаются 20 студентов. Занятия проводят квалифицированные преподаватели, многие из которых закончили высшие учебные заведения в СССР, НРБ, ВНР. На базе политехнического института прошли первичную подготовку преподаватели информатики.

Была поставлена задача создания необходимой материально-технической базы. В Монголии нача-

ты разработки компьютера «Образование» (8-рядный микрокомпьютер на базе микропроцессора К-580). А пока большую помощь МНР оказывают братские социалистические страны: функционируют два класса компьютеров «Правец-8М» (НРБ), приобретены учебные компьютеры «Препер-16» (ВНР).

В Политехническом институте проводится работа по созданию программного обеспечения для учебного процесса. Разработку программ ведут преподаватели и студенты института. На рабочем совещании заместитель заведующего кафедрой Политехнического института Д. Бадарч продемонстрировал некоторые из них. Хорошие программы по информатике и начертательной геометрии, их отличительная черта — широкое использование графических возможностей микрокомпьютера «Правец-8М». С 1987/88 учебного года в школах МНР началось изучение предмета «Информатика». Пока в основе курса советская программа и учебники, но уже ведется работа по созданию собственного оригинального учебника.

**М. ЛЕБЕДЕВА,**

зав. отделом ВНИИ профтехобразования,  
канд. пед. наук



## О печальном и забавном

Начнем, пожалуй, с письма, адресованного в Управление информатики и электронно-вычислительной техники бывшего Минпроса СССР и недавно переданного в редакцию.

126

*Я работаю учителем информатики второй год. В этом учебном году был оборудован межшкольный кабинет ВТ на базе БК-0010 и ДВК-2М. В своей работе встречаюсь с большими трудностями. Помощи, в отличие от учителей других дисциплин, получить негде. Мне самой приходится осваивать не только БК-0010, но и более сложную ДВК-2М. Математическое обеспечение отсутствует полностью. И, как с тревогой и озабоченностью констатирует журнал «Наука и жизнь» (№ 4, 1987 г.) в статье М. Максимова, написать программу на Фокале по силам далеко не каждому. А я вижу, как 7—8-классники с любопытством засматриваются на перерывах на исписанную доску и еще с большим интересом заглядывают в КВТ, и считаю своим первейшим долгом развить этот интерес, не дать затухнуть огоньку любознательности.*

*Вот и отправилась я в Москву. Посоветовали обратиться в коммерческое объединение «Электроника» (ул. Раменки, 12), где часовая консультация стоит 5 руб. Однако там Ольга Константиновна, побеседовав со мною около получаса, позвала на помощь Леонида Юрьевича. Консультация обошлась в 13 руб. 50 коп. Но, имея 6 машин ДВК-2М, так и не смогли прочитать содержание дисков, привезенных мною. Одним словом, свой «компьютерный багаж» пополнить так и не удалось. А с чем же вернуться к своим ученикам? Они, как и я, многого ждали от посещения столицы. В клубе «Контакт» при редакции журнала «Наука и жизнь» мне посоветовали обратиться на Смоленский бульвар, 4, где и приобрела я за 12 р. 25 к. одну кассету с 5 играми. Причем игру «Скачки» мы дома так и не разгадали. За консультациями по Фокалу обратилась в школу № 72, где проходили испытания КУВТ-86, к учителю Константину Александровичу Шульману. У него я получила практически все, во имя чего приехала в столицу. Однако применить его советы, практические рекомендации у себя на месте не могу, поскольку ДВК-2М, не проработав и часа, была отправлена в ремонт в ПО ВТИ, где находится до настоящего времени. Две БК-0010, не проработав и месяца, уже около двух месяцев находятся в ремонте. В ремонт требуется отправить еще две.*

*Но как работать с классом, если Павловский Посад не поставляет блоков в ремонтные объединения и все неработающие компьютеры (по словам ПО ВТИ) отвозят на ремонт изготовителям? Кажется, есть над чем подумать министерствам просвещения РСФСР, СССР и тем, кто занимается производством и поставкой в школы этих компьютеров.*

**В. СИДОРОВА,**  
г. Острогжск Воронежской обл.

Что тут скажешь? Компьютерный центр «Электроника» оказался не на высоте, а вот коллега-учитель сумел помочь. Пожалуй, это даже естественно — центр «Электроника» создан недавно, а лучшие учителя успели приобрести немалый опыт. Но вот почему из Острогжска нужно ехать за консультацией в Москву? Ведь в Воронеже хороший ИУУ! Там находится также одно из крупнейших производственных объединений «Электроника». Неужели нельзя организовать использование малой части его огромного потенциала в интересах школы?

Ремонт... Увы, эта тема не сходит с повестки дня. Недавно к ней обратилась «Правда». Заместитель директора завода «Экситон» из Павловского Посада обещает в нынешнем году повысить качество выпускаемой продукции и наладить снабжение ремонтников запчастями. Для этого завода придется организовать собственный участок по производству микросхем. Несколько странный путь, но если он улучшит положение — скажем «спасибо».

Кстати, совет «бывалого пользователя»: не пишите мелом на доске в кабинете вычислительной техники, если хотите, чтобы диски и дисководы служили долго. Меловая пыль очень опасна!

А теперь о некоторых недоразумениях, возникших между читателями и журналом.

Нас порой упрекают в недемократичности. «Мне кажется, вы поставили перед собой одну-единственную цель — благоприятствовать внедрению в наше образование идей академика Ершова А. П., его учебного курса и очень примитивного учебника» — пишет С. В. Богдашев из Великого Устюга и предлагает организовывать обмен точками зрения в рубрике «Точка зрения». А. Н. Векслера из Хабаровска не удовлетворяет наша молодежная рубрика: «Не лучше ли будет, если сами читатели ре-



шат, о чем будет информировать «Молодежная инициатива»?»

Лучше! И давайте обмениваться точками зрения! Ведь эти рубрики самими названиями своими говорят (может быть, достаточно ясно?), что созданы специально для проявления читательской инициативы и обмена разными мнениями (но это не означает, что в другие рубрики читателям «вход воспрещен»).

Для полной ясности, однако, следует заметить: мы хотим сделать «Точку зрения» трибуной для изложения самого широкого спектра взглядов, но это не значит, что в ней будет публиковаться всё подряд. С. В. Богдашев, например, предлагает (не аргументируя) изучать в VII—VIII классах работу на программируемых калькуляторах, а в X — работу с текстовыми редакторами, базами данных и т. п., хотя последняя тема явно легче для семиклассников, чем калькуляторы. Правда, изучать текстовые редакторы (особенно хорошие, «дружественные») в VII классе сейчас невозможно — они не реализованы на подавляющем большинстве КУВТов.

О подобных реальностях часто забывают наши корреспонденты, доходя порой в своих фантазиях до утопий. «Мне кажется... что информатике обучать надо человека со дня его рождения, и тогда страна не будет испытывать недостаток в системных программистах» (А. Вербицкий, г. Северодонец).

Другое часто встречающееся «недоразумение» — просьбы о публикации текстов программ. «Чаще печатайте... пакеты прикладных программ, которые можно использовать в управлении учебным заведением (СПТУ, СШ) применительно к ЭВМ ДВК» (А. Н. Крупин, г. Кызыл-Орда); «Меня очень заинтересовала разработка Л. Павлович и О. Павлович относительно применения ЭВМ при контроле усвоения знаний. Хотелось бы поближе познакомиться с этой работой. Не отказался бы познакомиться с самим текстом программы» (Н. Векслер).

Мы печатаем программы, но это вынужденная мера. Их следует распространять на магнитных носителях, на грампластинках, как-то еще, но не в виде текстов. Небольшие программы, демонстрирующие концептуальные новинки, в журнале приемлемы, но пакеты прикладных программ... Это не лучшее использование бумаги. И не только потому, что журнал может превратиться в издание для ЭВМ, а не для людей. ЭВМ требуют абсолютной точности, а людям свойственно ошибаться. Свежий печальный пример — статья Г. Тимошенко во втором номере за этот год. Авторские ошибки не были замечены рецензентом, редактор также не улучшил текст, свою лепту внесли и наборщики — и вот результат: многие программы не работают. К сожалению, несмотря на все усилия, от подобных случаев мы не застрахованы. Ведь даже распечатка, сделанная на принтере и воспроизведенная в журнале, может содержать авторские ошибки, а может — и привнесенные сбоем ЭВМ (как, например, в публикации Е. Гопергауза — № 5 за 1987 г.).

Очень много копий ломается в спорах о том, какой язык программирования лучше; основной камень преткновения — Бейсик. «Делаются попытки дискредитировать Бейсик — самый простой для освоения язык, версия MSX которого дает богатейшие возможности, особенно для создания обучаю-

щих программ» (С. В. Богдашев). Редакция обвивает в противодействии этому языку, то в «подсуживании» ему же... А повода для спора нет. Каждый язык хорош на своем месте и в свое время. Если на ПЭВМ есть только транслятор с Фокала, то придется работать на Фокале — и, право, можно сделать немало хорошего!

То, что мы печатаем высказывания, характеризующие позиции авторов по отношению к различным языкам программирования, не значит, что журнал особо поддерживает (или осуждает) какой-то из них. Выбирайте сами, если у вас есть выбор! Но остерегайтесь фанатизма.

М. Колесников из Белгорода прислал большое и в целом интересное письмо, однако трезвость подхода изменила ему, как только была затронута тема языков: «...язык академика Еришова значительно проигрывает в наглядности блок-схемам. Излишне усложнен. В нем вводятся такие элементы и конструкции, которых нет ни в одном «живом» языке. Например, цикл пока или условие если — то — иначе. Во всех языках, с которыми мне приходилось сталкиваться, действия иначе подразумеваются по умолчанию!»

Во-первых, конструкции, аналогичные алгоритмической нотации пробного учебного пособия, широко используются во всем мире.

Во-вторых, цикл пока и явное задание ветви иначе есть в Алголе, Фортране, Паскале, ПЛ/1, в некоторых версиях Бейсика...

Дорогие читатели! Гораздо продуктивнее отыскивать в различных языках программирования их хорошие черты, позволяющие с максимальной эффективностью решать определенные задачи, чем высказывать недостатки. Проверьте сами!

И еще о двух недоразумениях более локального характера. В. В. Ш. (внимательный и вдумчивый читатель, предпочитающий не раскрывать свое имя) пишет: «В информатике многие термины, конечно, не устоялись, но тем не менее непонятно, почему описываемая в статье А. Архангельского операционная система названа «девичьей» фамилией RT-11; у нас она известна под названием РАФОС». У нас различные ее версии и модификации известны также под названиями ОС ДВК, ФОДОС (с различными номерами), FOBOS, именно поэтому автор предпочел использовать оригинальное название ОС.

К. Г. Прошин из Перми спрашивает нас: «Правильно ли я понял, что можно приобрести в личное пользование ПЭВМ одного из типов (ПМ1, ..., ПМ5), на которые делятся ПЭВМ согласно ГОСТ 27201—87?»

Увы, Константин Георгиевич, вы поняли неправильно. О продаже за наличный расчет любых персональных компьютеров в заметке «Стандарт на ПЭВМ» речь не шла. Продаваться должны только ПЭВМ группы ПМ1, предназначенные специально для использования в быту. Есть надежда, что будут продаваться машины группы ПМ2; когда, сказать трудно.

Собственно, бытовые компьютеры (БК-0010, «Микроша») продаются уже сейчас, однако упомянутому ГОСТу они не соответствуют и потому не могут быть отнесены даже к группе ПМ1.



## Веселый урок



### Диалог

— Сережа, дорогой, как тебе нравится мое новое платье? Элегантное?

«Какой ужас! Элементарную программу не могу составить».

— Сергей, ты считаешь, что можно не отвечать мне? Какой ужас!

«Считаешь, считаешь — не получается. Что-то в программе».

— Я для тебя ноль...

«Неужели деление на ноль? Быть не может».

— Ты не можешь уделить мне одну минутку. Ты вообще не видишь меня, смотришь как через стекло.

«Причем здесь стек...»

— Всё, моя чаша терпения переполнилась. На переполнение не похоже».

— Я знала, что ошибаюсь, еще когда мы регистрировались.

«Ошибка в регистрах!.. Конечно, не из того беру».

— Ты никогда не понимал меня. Я измучилась.

«Всё понятно. Эх, если бы иметь машину, умеющую по-настоящему вести диалог с человеком, разве бы я мучился».

— Раз ты этого хочешь, я уйду. Ты не любишь меня. Всё. Кончено.

— Галинка, милая. Всё. Ура! Кончено. Готова программа. Теперь мы заживем.

«Не готова я уходить».

— Поторопился и ошибся. Еле исправил.

«Не стоит торопиться».

— Понимаешь, не из того регистра.

«И до регистрации он точно такой же был. Не говорит, не говорит, потом скажет слово, другое... Много ли женщине для счастья надо?»

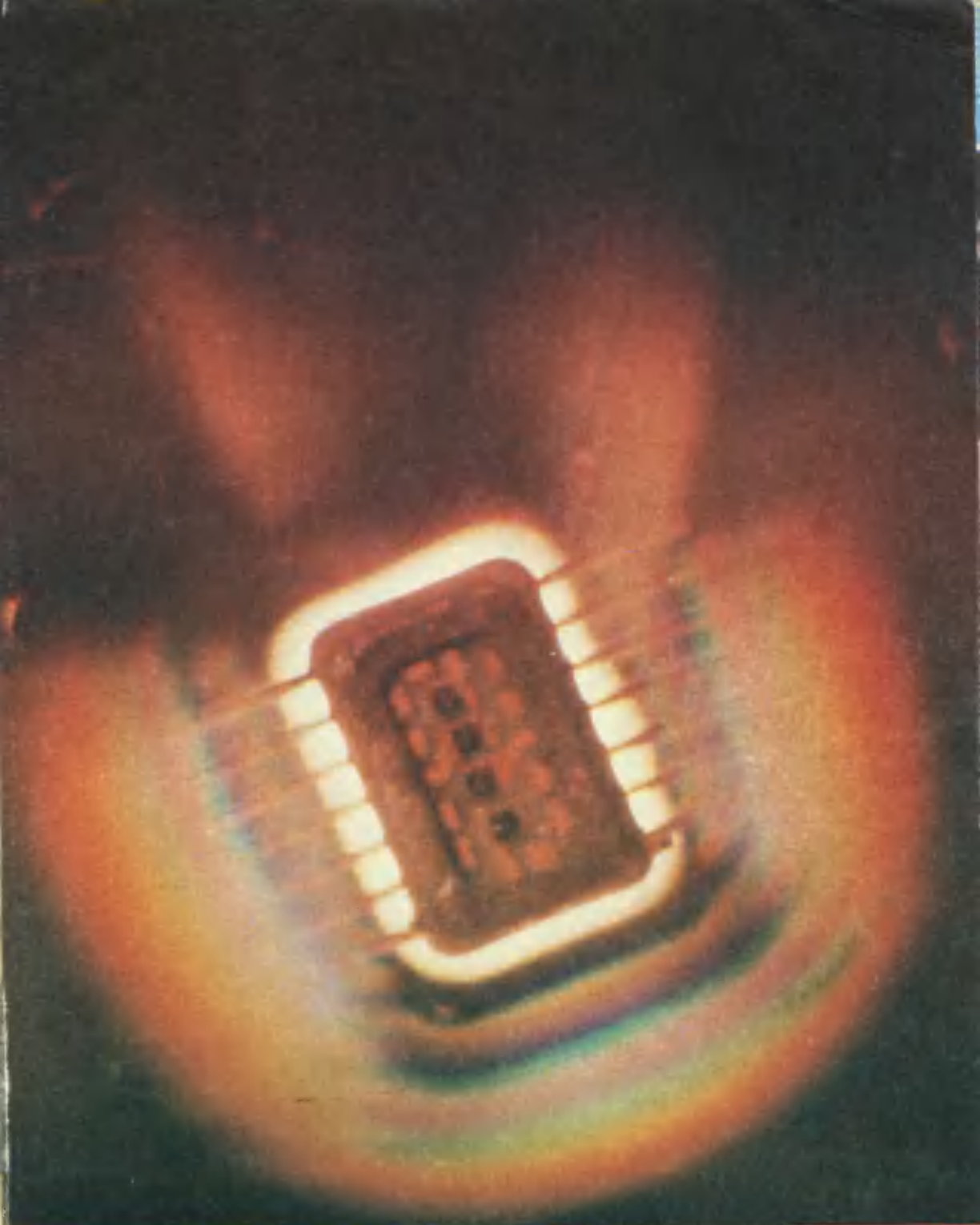
— Сережа, дорогой, как тебе нравится мое новое платье?..

### Поговорки

1. Семеро одного дисплея не ждут.
2. Перфокарте — место (из инструкции оператору).
3. Каждому программисту ЭВМ выдает то, чего он заслуживает (из опыта).
4. Что у заказчика на уме, то у программиста на языке.
5. Не спрашивай старого программиста, спрашивай бывалого.
6. Лучше с хорошим программистом метку потерять, чем с плохим найти.
7. Алгоритмический язык до Новосибирска доведет.
8. Хороша веревка длинная, а программа короткая.
9. Алгоритмам учиться — всегда пригодится.
10. Снявши оператор вывода, по ответу не плачут.
11. В чужой ВЦ со своим транслятором не ходят.
12. Массив не припасешь — памяти не будет.
13. Программу циклом не испортишь.
14. Ячейка память бережет.
15. С миру по ячейке — программисту банк данных.
16. Лучше микрокалькулятор в руках, чем персональный компьютер в мечтах.
17. Кто как программирует, так и ест.
18. Маленький алгоритм лучше большого безделья.
19. Отольются заказчику слезы программиста.
20. Дурной ФОРМАТ печати покоя не дает.

А. ПИСКУНОВ

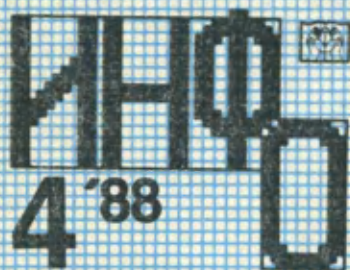
Б. МЕДВЕДЕВ,  
Ленинград



Кто силу знания отверг,— тот обделен умом,—  
Он счел, что сущее кругом — обманчивый фантом.  
Но если этот зримый мир — воображенья плод,  
Как с вечной истиною быть, вне нас живущей в нем?

Джами

Цена 60 коп.  
70423



**ИНФОРМАТИКА  
И ОБРАЗОВАНИЕ**

