

# ИНФОРМАТИКА И ОБРАЗОВАНИЕ

6 1989





«...На экспертизу в белибердоведческую лабораторию поступили две новые надписи. Предполагается, что одна из них содержит подлинный текст, а другая является подделкой...» (статья «Квазиязык на ЭВМ в учебных целях»).



# ИНФОРМАТИКА И ОБРАЗОВАНИЕ

OldPC.su

7 0 0 2

музей компьютеров

## Содержание

### Общие вопросы

Гриценко В., Довгялло А. Пути развития информатизации образования 3

### Методика обучения

Каймин В., Кугель Л., Кузницкий Е., Угринович Н. Алгоритмизация и начала программирования 13  
Кушниренко А. О новом учебнике 25  
Долматов В. Методические проблемы построения курса информатики для V—VI классов 34  
Савченко В. Как составить алгоритм 39  
Скобелев Г. ПМК на уроках математики 43

### КВТ

Гриценко А. Информационно-поисковая система для КУВТ-86 45  
Алексеев М., Алексеева Т. Твои первые программы 50  
Павлов С., Цилевич Б. ППС на основе математического моделирования 64  
Салимжанов Р. Стиль программирования 69  
Карпов В., Новичков В. Графика на алфавитно-цифровом дисплее: реализация на Фортране 73  
Гужа И. Расширение возможностей КУВТ-86 76  
Шмелев А. Генератор программ-тренажеров 78  
Рынок ЭВМ: Минрадиопром 79  
Клуб пользователей БК 82

### Педагогический опыт

Санжаров Л., Финьков А. Квазиязык на ЭВМ в учебных целях 85  
Симонов А., Сысоев И., Шахова Ю. Программа-тренажер по астрономии 87  
Коган А. Отчего кошку назвали кошкой? 89  
Кулова Л., Мовсесян Ю. Как учить информатике сегодня? 90

### Педагогические кадры

Зеленин В. Метод проектирования ППС по структурированным образцам на языке MSX-Бейсик 93  
Ефимов В. Компьютер в педвузе 96

## Внеклассная работа

Рсвунов А. Школьники выбирают дорогу  
Олимпиады по информатике 99  
100

## Точка зрения

Волков Д. Школа, программирование, познание 104

## Интервью номера

Я выбрал систему профтехобразования 108

## Достижения и перспективы

Экология мышления 112

## Библиография

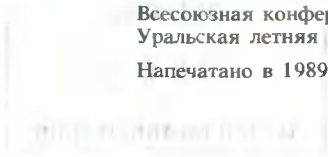
По страницам тематических планов издательств на 1990 г. 118

## Информация

Всесоюзная конференция в Севастополе 123

Уральская летняя школа 124

Напечатано в 1989 г. 126



Обложка Э. Бажилина

В оформлении номера принимали участие: Э. Бажилин, Н. Горбунова,  
С. Расторгуев

Главный редактор  
академик  
В. А. МЕЛЬНИКОВ

Редакционная  
коллегия  
И. Н. АНТИПОВ  
В. Н. АФНАСЬЕВ  
И. М. БОБКО  
Г. В. ГОДЖЕЛЛО  
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  
К. В. ШЕХОВЦЕВ  
(редактор отдела)

Редактор отдела А. Кравцова  
Научный редактор Т. Драгныш  
Зав. редакцией Н. Игнатова  
Художественный редактор Л. Розанова  
Корректор О. Пурлова

Сдано в набор 20.09.89. Пдписано в печать 31.10.89. А 03565 Формат 70×100/16.  
Бумага офсетная № 1. Печать офсетная. Усл. печ. л. 10,40. Усл.-кр. отт. 42,88.  
Уч.-изд. л. 13,67. Тираж 83515 экз. Заказ 2277. Цена 60 коп.

Издательство «Педагогика» Академии педагогических наук СССР  
и Государственного комитета СССР по печати.

Адрес для переписки: 107005, Москва, Лефортовский пер., 8.  
Адрес редакции: Студенческая ул, 37.  
Телефон редакции: 249-97-77.

Ордена Трудового Красного Знамени Чеховский полиграфический  
комбинат Государственного комитета СССР по печати.  
142300, г. Чехов Московской обл.

© «Педагогика», «Информатика и образование», 1989

В. ГРИЦЕНКО, А. ДОВГЯЛЛО

## Пути развития информатизации образования

Утверждается, что направленное и расширяющееся воздействие информатики, микроэлектроники и компьютерной техники на образование<sup>1</sup>, науку и технологию, культуру и коммуникации представляет собой одну из важнейших междисциплинарных проблем и вызовов нашего времени...

*Из резолюции 25-й сессии  
Генеральной конференции  
ЮНЕСКО [1]*

### I. Введение. Общие требования к информатизации образования

Новые информационные технологии (НИТ) становятся основной движущей силой развития. «Их правильное или неправильное применение серьезно отражается на конкурентоспособности сельского хозяйства и промышленности, эффективности сферы услуг, на стоимости и качестве образования, обороноспособности, способности к нововведениям как в экономических, так и в социальных областях» [2].

Как следствие усложнения и быстрого видоизменения технологий, непрерывно увеличивается объем и изменяется содержание знаний, умений и навыков, которыми должны владеть современные специалисты, все члены информатизируемого общества. Во всех сферах

образования и подготовки кадров в национальном, региональном и международном масштабах ведутся поиски способов интенсификации и быстрой модернизации системы образования, повышения качества обучения путем применения компьютеров для поддержки самостоятельной познавательной деятельности обучаемых различных категорий, интенсификации труда преподавателей, управления работой учебных заведений.

Особую остроту приобретает вопрос непрерывной опережающей подготовки и переподготовки специалистов самых различных категорий к эффективному использованию НИТ, современных компьютеров в своей деятельности. Умение пользоваться вычислительной техникой при решении профессиональных и учебных задач по праву приравнивается сейчас ко второй грамотности. От ее уровня и распространения прямо зависят развитие научно-технического прогресса, интенсификация, модернизация и интеллектуализация производства и самой системы образования.

Требования к информатизации образования могут быть различными в зависимости от общей концепции развития мирового сообщества (см., например, [3]). В рамках одной из них «быть наилучшим» означает быть самым конкурентоспособным в плане экономической рентабельности. Отсюда основной лозунг: «занятость для наилучших».

<sup>1</sup> Подчеркнуто нами.— Примеч. автора.

Другие будут иметь доступ лишь к весьма нестабильной, низкооплачиваемой работе либо будут получать государственную помощь по бедности.

Другая концепция в качестве ключевых слов для описания нового общества предлагает следующее: «повышать уровень жизни» и «проявлять самоубытность и солидарность», что подразумевает возможность производственного обучения и приобретения квалификации каждому человеку и каждой социальной группе, а также самый широкий доступ к основным материальным ценностям и услугам в интересах подавляющего большинства. Вторая концепция содержит в себе логику сотрудничества, т. е. использования самых различных механизмов, чтобы выявить оптимальные интересы каждого благодаря объединению ресурсов, знаний и опыта всех членов общества. В рамках второй концепции людские ресурсы изменяют свой статус: они больше не являются «сырьем», одним из факторов производства, а становятся субъектом процесса организации и овладения факторами материального и нематериального производства.

Не будем отдавать явного предпочтения одной из них, но сформулируем общие требования (цели) информатизации образования с учетом обеих.

1. Доступность знаний и данных для каждого члена общества.

2. Развитие интеллектуальных и творческих способностей индивидуума (с учетом его самостоятельности, самоубытности и т. п.).

3. Сотрудничество (обмен, солидарность).

4. Непрерывное повышение квалификации или изменение области профессиональной деятельности в течение жизни каждого члена общества.

5. Гуманизация общего образования и воспитания.

6. Опережающее образование.

7. Обучающее сопровождение информационных технологий, всеобщая компьютерная грамотность.

8. Интенсификация обучения и образования.

Информатизации образования предшествуют другие тенденции его модернизации. Первые попытки автоматизации и электронизации<sup>2</sup> образования в СССР относятся к 1962 г., когда академик А. И. Берг организовал проведение работ по программированному обучению, ТСО и обучающим машинам. В том же 1962 г. академик В. М. Глушков на первой Всесоюзной конференции по программированному обучению поставил задачу разработки компьютерных методов и средств для автоматизации обучения (рис. 1).

В последующие годы в СССР вернулись работы по созданию автоматизированных обучающих систем (АОС) первого поколения (на машинах Днепр-21, Минск-32 и др.), а также АОС второго поколения, базирующихся на специальных инструментальных средствах — пакетах прикладных программ типа СПОК, АОС-ВУЗ [6, 11, 16]. Ведущую роль в координации этих работ сыграли Институт кибернетики АН УССР и НИИ проблем высшей школы Минвуза СССР.

Активизация работ по АОС в СССР приходится на первую половину 80-х гг., когда прикладные и инструментальные учебные средства на базе семейства типовых ППП АОС-ВУЗ (ЕС-ЭВМ) получили довольно широкое распространение в учебных заведениях страны и за рубежом (в НРБ и ГДР). В 1985 г. были приняты известные постановления КПСС и правительства СССР о компьютеризации средней и высшей школы, создавшие новую идеологическую и материально-техни-

<sup>2</sup> Современное представление об электронизации образования включает в себя три основных аспекта: 1) широкое обучение электронике и применению средств микропроцессорной техники; 2) интенсивное использование в учебном процессе технических средств обучения (ТСО), включая обучающие машины, видеосистемы, простые электронные игры и тренажеры; 3) внедрение в учебный процесс микрокалькуляторов, специализированных учебных ЭВМ и др. (Цага З. История и современность электронизации в воспитании и образовании в ЧССР. Прага: ЧВУТ, 1988).

ческую основу для компьютеризации системы образования.

Выполняемую в настоящее время программу компьютеризации образования следует рассматривать как неотъемлемую составную часть общенациональной программы повсеместного внедрения достижений информатики и вычислительной техники и подготовки общества к использованию этих достижений. Следовательно:

компьютеризация основных уровней подготовки кадров для научно-производственной деятельности (общеобразовательная и профессиональная школа — вуз — производство/наука) должна быть увязана единством подхода, методов, целей;

средняя школа должна заложить основу дальнейшего совершенствования знаний и умений на всех уровнях;

с целью интенсификации процесса обучения и широкой пропаганды достижений информатики и вычислительной техники необходимо обеспечить внедрение на всех уровнях образования автоматизированных учебных курсов (АУК), реализующих компьютерное представление знаний не только в области информатики и вычислительной техники, но и в других предметных областях — математике, физике, инженерном деле, гуманитарных науках.

Качественно новый этап работ в компьютеризации образования связан со становлением учебной информатики [10, 18, 23], основанной на единой для всех уровней системы образования компьютерной технологии обучения

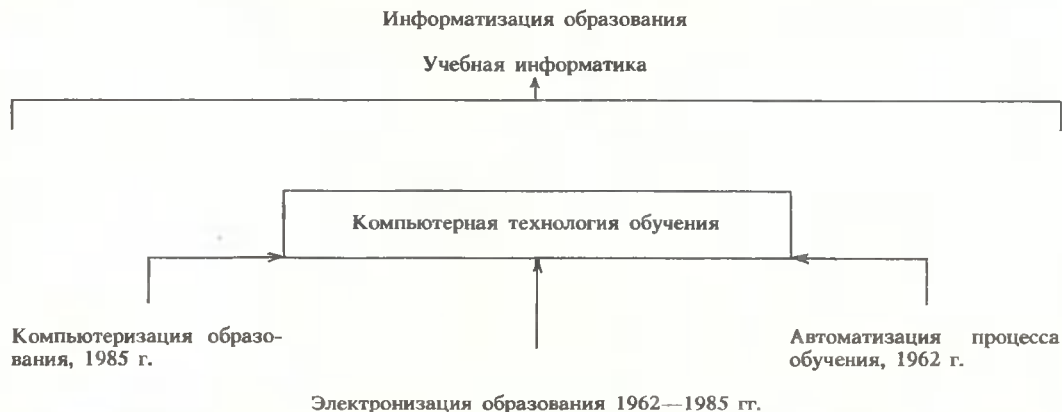
(КТО), т. е. технологии проектирования, программирования и тиражирования (внедрения) машинных знаний учебного назначения, реализуемых на базе персональных ЭВМ, систем коллективного пользования и сетей ЭВМ и обеспечивающих кооперативный и игровой режим совместного человеко-машинного решения задач, поддержанный автоматизированным обучением.

КТО должна быть основана на формализации знаний и учебных задач, включении в обучающий диалог средств и методов искусственного интеллекта и на применении для эффективной реализации сценариев обучающего взаимодействия семейства специальных пакетов прикладных программ, в том числе и ориентированных на преподавателя-непрограммиста.

5

### III. Компьютерная технология обучения как основа информатизации образования

Как и компьютеризация образования, информатизация [4, 5] предполагает преобразование комплекса средств и социальных условий развертывания информационных процессов во всех сферах общественной деятельности на новой технико-технологической, организационной и юридической основе, включающей ЭВМ, системы компьютерных коммуникаций, банки данных и знаний, робототехнические комплексы, средства сбора, ввода, хранения, отображения и представления информации при условии организации эффективного взаимо-



Электронизация образования 1962—1985 гг.

Рис. 1. Исторические предпосылки информатизации образования в СССР

действия между указанными средствами и человеком.

Исходя из вышеизложенного, концепцию информатизации образования сформулируем следующим образом: компьютерная грамотность, информационная оснащенность и непрерывное опережающее преемственное индивидуализированное развивающее образование и воспитание каждого члена общества.

Компьютерная технология обучения, как основа информатизации образования, направлена на достижение указанных выше целей на основе применения комплекса функционально зависимых педагогических, информационных, методологических, психофизиологических и эргономических средств и методик, созданных и организованных на базе технического и программного обеспечения ЭВМ.

КТО предназначена для обеспечения жизненного цикла (включая создание, применение или эксплуатацию и развитие) информационных прикладных и инструментальных средств в виде учебно-технологических модулей (УТМ), обеспечивающих эффективную деятельность обучаемых и преподавателей на уроках, лекциях, семинарах, практических и лабораторных занятиях, зачетах и экзаменах, при самостоятельной работе обучаемых, курсовом и

дипломном проектировании, других видах познавательной деятельности.

Сравнительно с работами, которые в прошлом были направлены на модернизацию процесса обучения, современный подход<sup>3</sup> к КТО строится на качественно новой концепции — концепции комплексного прагматического представления и поэтапной детализации и активизации знаний учебного назначения [10, 12, 23].

В чем суть прагматического подхода к системам представления и обработки знаний, прежде всего знаний учебного назначения?

В последние годы представление знаний с помощью ЭВМ ассоциируется главным образом с системами искусственного интеллекта и экспертными системами. Прагматический подход к этому новому и пока еще дискуссионному направлению рассматривает компьютерное представление знаний с более широкой точки зрения, при которой материальной формой знаний учебного назначения может быть и текстовый и графический материал на машинных

<sup>3</sup> Речь идет о подходе, развиваемом Институтом кибернетики им. В. М. Глушкова АН УССР, НИИ проблем высшей школы ГКНУ СССР и организациями, участвующими в работах по проблемам 1.1.6 и 1.2.7 КП НТП СЭВ 6÷12.

#### Классы средств представления знаний

##### Прикладные средства

6. Экспертно-обучающие системы<sup>4</sup>, интеллектуальные обучающие системы
5. Решатели задач, экспертные системы
4. Тренирующие и диалоговые системы, учебные игры
3. Автоматизированные обучающие и контролируемые системы
2. Информационно-справочные системы
1. Встроенная документация, справочные и другие учебные тексты
0. Печатные инструкции для пользователя

##### Инструментальные средства

«Пустая» ЭОС<sup>4</sup>

Инструментальные средства обработки знаний (типа Интерэксперт, Пролог, ЕЯ лингвистические процессоры)  
Интегрированные пакеты прикладных программ (типа АОС/РРВ, АОС-ВУЗ/ПРИМУС); интерактивная графика  
Пакеты прикладных программ учебного назначения (типа ТЬЮТОР, КУРСРАЙТЕР, АОС/ВУЗ)  
Система управления базами данных (СУБД)  
Текстовые и графические редакторы

Типографские машины

Детализация и активизация знаний

Рис. 2. Эволюция средств представления знаний (прагматический подход)

<sup>4</sup> Экспертно-обучающая система (ЭОС) — совокупность взаимодействующих экспертных систем по данной предметной области, методике обучения, диагностике и объяснению ошибок обучаемого [12].



### Компьютерная технология обучения

Прикладные УТМ		Инструментальные УТМ			
		Спец. системы программирования	Авторские системы	Генераторы прикладных УТМ	
Обучение	Учение	Справочный и другие учебные тексты, встроенная документация Системы демонстрации слайд-фильмов Информационно-справочные системы	Я А	Э Л Б К О Н АСКУН	
	Контроль знаний	Средства контроля знаний	0		
	Тренировка	Обучающе-контролирующие курсы и АОС	К В	РАКУРС	
		Интегрированные АОС Тренажеры Учебные игры	С		
	Решение задач	Имитационные (моделирующие) системы ЕЯ вопросно-ответные системы		ЗАПСИБ-ПРОЛОГ; КЕУ ИНТЕРЭКСПЕРТ	
	Диалоговое решение теорет. задач	Решатели задач, экспертные системы	П В Р Е С О S Л S Л О Г		
		Экспертно-обучающие системы			
	Диалоговое решение практ. задач	Учебные роботы и ГАПы	П	ЭЛЬКОН/Р	

Рис. 3. Программные средства КТО

носителях, и информационно-справочная или контролирующая система, и традиционная сценарная АОС по какому-либо предмету. В качестве доказательства правомерности такого подхода на рис. 2 дана в несколько упрощенном виде эволюция средств представления знаний о компьютере. Аналогичная картина может иметь место и для других предметных областей (медицины, биологии, математики), естественно, при наличии соответствующих инструментальных программных средств.

В компьютерной технологии обучения прикладные средства представления знаний учебного назначения должны быть реализованы в виде прикладных программно-технических УТМ, систематизация которых (в порядке их усложнения) дана на рис. 3. Слева перечислены дидактические функции, которые должны обеспечиваться прикладными УТМ.

Автоматизированный учебный курс по какой-либо предметной области — это организованная совокупность прикладных УТМ. В деле создания АУК прагматический подход к представлению и обработке знаний учебного назначения выражается в следующем:

для преподавателя обеспечивается возможность компоновки конкретного АУК (задаваемого целью обучения, учебным планом проведения занятия и контингентом обучаемых) из набора прикладных УТМ;

для автора АУК предоставляется возможность эффективно использовать как отдельный инструментальный УТМ, так и любое их сочетание.

КТО предусматривает эффективное создание программных и информационных средств учебного назначения как программистами, так и лицами, не имеющими специальной подготовки в области программирования. Для под-

держки их деятельности инструментарий КТО должен включать (см. рис. 3):

специализированные системы программирования знаний учебного назначения (типа, например, АОС/ВУЗ, АОС/М [11, 16], АВС<sup>5</sup>, ТУРБО-ПРОЛОГ, СЛП<sup>5</sup>);

авторские системы, обеспечивающие создание конкретного прикладного УТМ по его сценарию или проекту в режиме «вопрос — ответ автора курса» (типа, например, ЭЛЬКОН<sup>5</sup>, АСКУН<sup>5</sup>, РАКУРС, ИНТЕРЭКСПЕРТ, ЗАПСИБ-ПРОЛОГ [7], KEY<sup>5</sup>);

генераторы прикладных УТМ по их неформальной спецификации (в перспективе).

Значительно меньше можно сказать в настоящее время об инструментальных средствах КТО, обеспечивающих автоматизированную детализацию и активизацию знаний от их текстового представления до базы знаний ЭОС (см. рис. 2). Эти инструменты должны обеспечивать автоматизированное преобразование текстовых файлов в базы данных ИСС, в сценарные АОС и, наконец, в базы знаний экспертных и экспертно-обучающих систем, т. е., образно выражаясь, обеспечивать ускоренную эволюцию прикладных средств представления знаний учебного назначения, представленных на рис. 2. Более подробно методология поэтапной детализации и активизации знаний изложена в [10, 12].

Каким образом способствует КТО достижению названных выше основных целей информатизации образования? Приведем некоторые примеры.

Создание и повсеместное распространение средств поддержки учения (справочных текстовых и графических материалов, слайдовых учебников и лекций, ИСС учебного назначения, имитационных моделей и учебных игр) на базе ПЭВМ, локальных и распределенных сетей ПЭВМ обеспечит доступность знаний и данных для каждого члена общества и возможность непрерывного повышения квалификации.

Диалоговые (в смысле [7, 8]) АОС, тренажеры и ЭОС — основа для индивидуализированного опережающего обуче-

ния и развития творческих и профессиональных способностей обучаемых самых различных категорий (от младших школьников до пожилых профессоров-гуманитариев, применяющих КТО в своей деятельности).

Экспертные системы, диалоговые решатели задач, учебные игры с многими участниками — основа для выработки склонностей и навыков к солидарности, сотрудничеству и обмену.

Обнащение программных продуктов и перспективных информационных технологий автоматизированными учебными курсами — путь к обеспечению высококачественной компьютерной грамотности, к непрерывному повышению квалификации всех членов информатизированного общества.

И наконец — краткая информация об интенсификации обучения и образования на основе применения средств КТО, с которыми обучаемый может работать как самостоятельно, так и в условиях учебной группы, класса. Как показывают имеющиеся экспериментальные данные, применение АОС, средств контроля знаний в 1,5—2 раза сокращает время и повышает качество усвоения многих предметов по сравнению с традиционной формой обучения. При этом резко возрастают активность обучаемых, количество решаемых ими учебных задач. КТО освобождает преподавателя от изложения и отработки с обучаемым значительной части учебного материала. Высвободившееся время преподаватель использует для дополнительной индивидуальной работы с отстающими или для постановки успевающим новых интересных заданий. Такое видоизменение работы педагога в условиях классно-урочной работы с применением средств КТО улучшает познавательную деятельность всей учебной группы в целом, усиливает творческие компоненты труда преподавателя.

Интенсивная работа обучаемого и преподавателя в условиях информационного комфорта, обеспечиваемая средствами КТО, приведет, в конечном счете, к гармоничному комплексному достижению всех целей информатизации образования.

<sup>5</sup> Новые разработки ИК АН УССР.

#### IV. Проблемы и пути информатизации образования. Оценка достигнутых результатов

Из всего спектра социальных и экономических проблем информатизации образования выделим проблемы создания и внедрения КТО как первоочередные и практически реальные.

Первая проблема относится к стадии проектирования режимов взаимодействия пользователя и ЭВМ при учении, обучении, диалоге и др. Она состоит в разработке теоретического аппарата для представления человеко-машинных алгоритмов совместного решения прикладных и учебных задач пользователем и системой.

Вторая проблема относится к стадии программирования спроектированных алгоритмов. Решение ее состоит в создании семейства взаимосвязанных мобильных инструментальных учебно-технологических модулей и линий для представления знаний учебного назначения в духе изложенного здесь прагматического подхода к представлению знаний.

Наконец, третья проблема состоит в тиражировании и внедрении («встраивании») средств КТО. Один из ключевых вопросов — опережающая подготовка профессорско-преподавательского состава в области КТО, создание сети автоматизированных учебных центров по КТО [9].

Важнейшим общим вопросом при решении всех этих проблем в региональном и международном масштабе является разработка унифицированной концептуальной и теоретической основы КТО (например, энциклопедического справочника и, в перспективе, базы знаний по КТО).

Что сделано к настоящему времени для решения указанных проблем и вопросов?

В последние годы в СССР, НРБ, ВНР и ГДР в качестве теоретической основы проектирования сценариев автоматизированного и диалогового обучения, других человеко-машинных алгоритмов все чаще начинают применять аппарат теории задач и способов их решения, разработанный в Институте

кибернетики им. В. М. Глушкова АН УССР при участии сотрудников института Психологии Минпроса УССР. Методической предпосылкой к разработке этого аппарата послужила идея выделить «задачные» модели основных структур единиц взаимодействия пользователя-обучаемого и ЭВМ и представить каждый конкретный сценарий обучения, диалога, игры и др. как процесс обработки этих «задачных» моделей двумя решающими системами — человеком и компьютером [7, 8].

Получает распространение методология структурирования баз знаний по конкретной предметной области и по так называемому дидактическому программированию, развиваемая в НИИ проблем высшей школы ГКНО СССР [6].

В плане решения проблемы инструментальной поддержки компьютерной технологии обучения, обеспечивающей эффективное программирование человеко-машинных алгоритмов и сценариев диалога, а также создание баз знаний по конкретной предметной области, основная цель состоит в разработке и всестороннем практическом опробовании развивающегося семейства совместимых специальных пакетов прикладных программ (СППП). Эти СППП предназначены для создания (см. рис. 3) информационно-справочных систем учебного назначения, сценарных АОС, интегрированных АОС (например, тренажеров по языкам программирования), а также для моделирования компонентов экспертно-обучающих систем.

Семейство СППП для АОС и некоторого класса диалоговых систем — явление новое для мировой практики. До сих пор разработчики подобных СППП ограничивались 1—2 пакетами, которые после морального старения обычно погибали вместе со всем внешним «слоем» прикладных обучающих, контролирующих, справочных программ. Пропадал многолетний труд десятков и сотен авторов машинных знаний в продуктивной сценарной форме представления.

Указанные методы, средства и системы прошли апробацию и/или внедрены в более чем 300 организациях СССР — вузах, институтах повышения

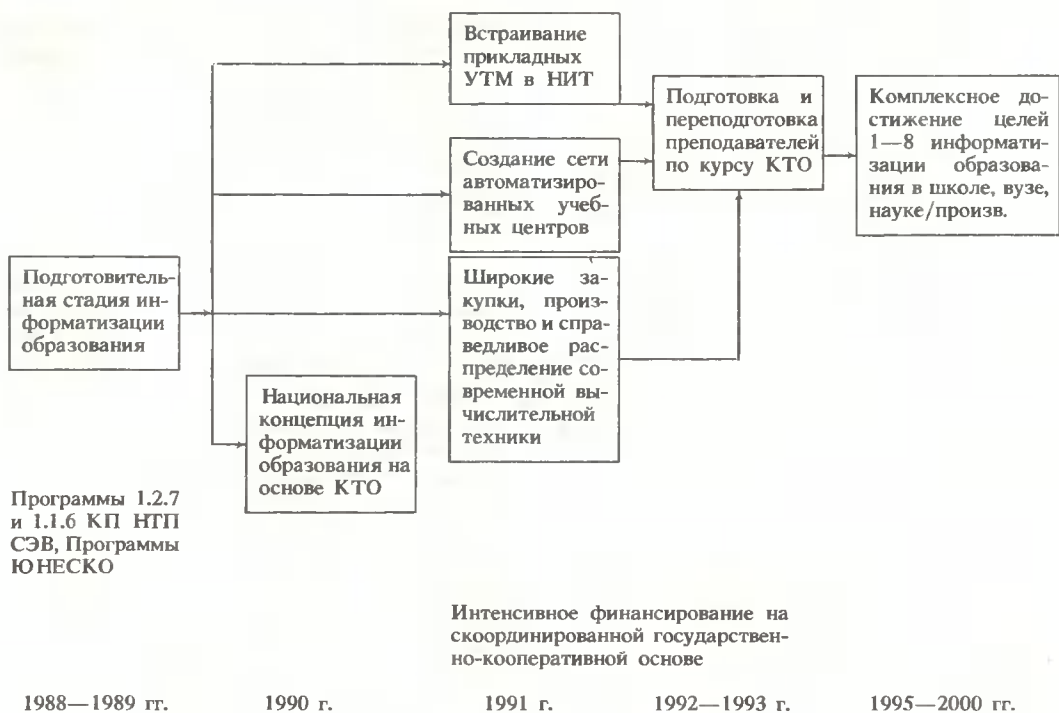


Рис. 4. Стратегия информатизации образования в СССР

квалификации, средних школах, в ряде организаций стран — членов СЭВ, в работах, выполненных по контрактам с ЮНЕСКО.

Создан и успешно функционирует при ИК АН УССР первый в стране автоматизированный учебный центр (АУЦ) информатики и компьютерной технологии обучения, располагающий уникальным набором прикладных и инструментальных учебно-технологических модулей КТО, демонстрационно-обучающими комплексами новых информационных технологий. Планируются работы по созданию базы знаний для генерации проектов профильных и региональных АУЦ.

В то же время КТО переживает сейчас противоречивый период своего развития. С одной стороны, начиная с 1985 г. эта технология (и прежде всего сценарные АОС, учебные игры, редкие пока экспертные системы учебного назначения) завоевала значительное «место под солнцем» индустрии информатики. С другой стороны, сейчас отмечается определенный «кризис жанра» и пересмотр основ компьютериза-

ции образования из-за:

большой трудоемкости изготовления эффективных и интересных АУК;

низкого уровня или отсутствия прикладных психолого-педагогических исследований, направленных на «встраивание» КТО в различные сферы образования;

слабого освоения и интеграции в функционально-единую технологию новых возможностей современных ПЭВМ.

Преодоление этих противоречий — предмет многочисленных международных и региональных конференций, симпозиумов и совещаний, количество которых экспоненциально растет в последние годы.

Проблема информатизации образования в настоящее время вышла за рамки национальной или даже региональной ее постановки и превратилась в международную проблему, решаемую всем мировым сообществом на новом качественном и количественном уровне.

Качественном — поскольку в научном плане проходит перестройка концепций автоматизации, электронизации и

компьютеризации образования. Сейчас ставится задача создания и внедрения КТО, основанной на изложенном здесь комплексном прагматическом подходе к представлению и обработке знаний.

Новый количественный уровень определяется переходом от работ в области компьютерной технологии обучения в рамках программ ГКНТ, АН СССР, ГКНО СССР к работам по программам ЮНЕСКО и по Комплексной программе научно-технического прогресса стран — членов СЭВ, включающей прежде всего программы 1.2.7 «Совершенствование системы образования на основе применения средств вычислительной техники» и 1.1.6 «Развитие технологии разработки и промышленного производства программных средств вычислительной техники».

Апробация изложенной здесь концепции информатизации образования в рамках указанных программ в системе высшего образования и при подготовке преподавателей в области компьютерной технологии обучения позволит поставить вопрос о создании и глобальном внедрении средств информатизации за всех уровнях образования.

## V. Первоочередные мероприятия по информатизации образования

Как следует из вышеизложенного, эффективная программа информатизации образования в национальном масштабе (рис. 4) должна включать следующие первоочередные мероприятия (выполняются согласованно и параллельно):

1. Выработка на конкурсной основе национальной концепции информатизации образования (1—1,5 года).

Необходимые предпосылки:  
новое понимание<sup>6</sup> компьютерной грамотности как моделирования объектов и процессов реального мира в компьюте-

ризованной лаборатории, а также ответственность подготовки в области информатики в школе, вузе, производстве/науке;

компьютерная технология обучения как основа информатизации образования.

2. Преодоление острого дефицита надежной современной вычислительной техники в учебных заведениях, включая школы и институты повышения квалификации (2—3 года)<sup>7</sup>.

Справедливое и равномерное распределение средств информатики и ВТ в стране.

3. Многоуровневая интенсивная подготовка и переподготовка всего профессорско-преподавательского состава по курсу «Компьютерная технология обучения» (см., например, [18]).

Информатизация образования требует развития профессионального мастерства преподавателей в рамках существенно модернизированной (главным образом за счет создания сети автоматизированных учебных центров) системы повышения квалификации.

Роль преподавателя в условиях внедрения КТО изменяется:

увеличивается объем учебной нагрузки на практических, семинарских, лабораторных и других видах занятий, относящихся к активным формам учебной деятельности;

самостоятельная работа студентов становится одним из основных видов учебной деятельности преподавателя;

интеграция научной и учебной работы требует от преподавателя дополнительных усилий, связанной с организацией межпредметных связей;

широкое информационное обеспечение учебного процесса на основе регулярного пополнения баз данных и знаний потребует от преподавателя по-

<sup>6</sup> Такой подход развивается в ФРГ, в частности, под руководством проф. В. Клауса. Результативность подхода была убедительно продемонстрирована на международных соревнованиях школьников по программированию (София, май 1987 г., Вторая международная конференция «Дети в мире информатики»), где западногерманские школьники получили 3 приза из 8.

<sup>7</sup> «За рубежом требуется 2—3 года, чтобы удовлетворить рынок новейшей вычислительной техникой. Там созданы и продаются высококачественные персональные компьютеры. А мы разве не способны это сделать? Ведь все это кому-то поручено и за все это нужно спросить» (из выступления депутата А. А. Логунова на II сессии Верховного Совета СССР // Московская правда, 1988. 30 окт.).

стоянного обновления и модернизации учебного материала.

Первоочередные работы:

разработка совместной для СЭВ учебной программы по курсу компьютерной технологии обучения для преподавателей;

создание унифицированного мобильного комплекса инструментальных и прикладных УТМ (включая электронную энциклопедию по КТО), поддерживающих указанный учебный курс, прежде всего в части интенсификации практических и лабораторных занятий.

4. Разработка программы интенсивного финансирования мероприятий 1, 2 и 3 в рамках скоординированных хозяйственных комплексов «государственное предприятие — кооператив».

## VI. Заключение

В эпиграфе к данному разделу образование не случайно, на наш взгляд, находится впереди всех других сфер информатизации. Действительно, многие специалисты и руководители, общественность в развитых и развивающихся странах высказывают убеждение в том, что информатизация образования (как эволюционный шаг в его автоматизации, электронизации, компьютеризации) является главным условием успешной информатизации общества.

Изложенная здесь концепция информатизации образования на основе КТО представляет, на наш взгляд, предпосылку к новаторскому проекту в области развития системы образования, органически сочетающему достигнутые результаты с образовательными и воспитательными потребностями развития мирового сообщества, наращиванием интеллектуальных возможностей общества, радикальным усовершенствованием советского образа жизни.

## Литература

1. Records of the General Twenty First Session; / Belgrade, 1980, 23 September to 28 October. Vol. 1. Resolutions, p. 115. Paris: / UNESCO, 1980.

2. Информатика в целях развития. Направления деятельности международной программы по информатике. Париж: ЮНЕСКО, 1985.

3. Петрелла Р. Человек в меняющемся мире // За рубежом. 1988. № 2.

4. Гриценко В. И., Панин В. Н. Информационная технология, вопросы развития и применения. Киев: Наук. думка, 1988.

5. Каныгин Ю. М. Индустрия информатики. Киев: Техника, 1987.

6. Савельев А. Я., Лобанов Ю. И., Новиков В. А. Подготовка информации для автоматизированных обучающих систем. М.: Высш. шк., 1986.

7. Диалоговые системы. Современное состояние и перспективы развития / Под ред. А. М. Довгялло. Киев: Наук. думка, 1987.

8. Довгялло А. М. Диалог пользователя и ЭВМ. Основы проектирования и реализации. Киев: Наук. думка, 1981.

9. Михалевиц В. С., Гриценко В. И., Довгялло А. М. Основные направления автоматизированного обучения кадров в области кибернетики, информатики и вычислительной техники // УСиМ. 1985. № 6.

10. Довгялло А. М., Никулин В. Н., Петрушин В. А. Учебная информатика и компьютерная технология обучения // Теоретические и учебные аспекты информатики. Киев: ИК АН УССР, 1987.

11. Автоматизированные обучающие системы на базе СПОК / П. Дановски, А. М. Довгялло, К. Н. Кирова и др. // Современ. высш. шк. 1983. № 1.

12. Экспертно-обучающие системы в комплексе компьютерных средств обучения / В. С. Михалевиц, А. Я. Савельев, А. М. Довгялло, Н. М. Когдов // Современ. высш. шк. 1988. № 1.

13. Harmon P., King D. Expert System: Artificial Intelligence in Business. New York: A. Wiley Pressbook, 1985. 289 p.

14. Балл Г. А. Базовые понятия общей теории задач. Киев, 1979 (Препр. / АН УССР, Ин-т кибернетики).

15. Представление знаний в АОС / Ю. И. Лобанов, В. В. Съедин, Н. Л. Васильев и др. М.: НИИ ВШ, 1985.

16. Новиков В. А. Типовые пакеты прикладных программ для автоматизированных обучающих систем. М.: НИИ ВШ, 1985.

17. Белов В. Н., Брановицкий В. И., Довгялло А. М. и др. Пролог — язык логического программирования / Прикладная информатика. 1986. Вып. 1.

18. Рабочая программа по курсу «Основы учебной информатики и вычислительной техники» для ФПК институтов иностранных языков / Под ред. А. М. Довгялло и П. И. Сердюкова. Киев: Минпрос УССР, 1987.

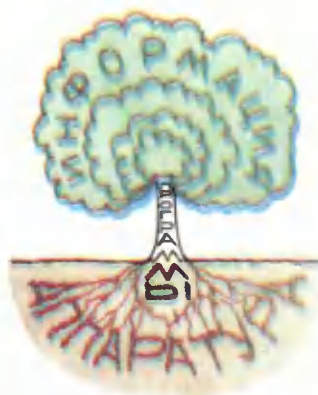
19. Стефик М., Эйкинс Я., Балзер Р. и др. Организация экспертных систем / Кибернет. сб. 1985. № 2.

20. Национальный прогноз развития компьютерной технологии обучения в СССР / Под ред. А. Я. Савельева. М.: ГКНО СССР НИИ ВШ, 1988.

21. Концепция информатизации образования / Под ред. А. П. Ершова. М.: ГКНО СССР, 1988.

22. Велхов Е. П. Опережающее образование // Экономическая газета. 1988. № 10.

23. Гриценко В. И., Довгялло А. М. Учебная информатика, компьютерная технология обучения и опережающая подготовка кадров // Материалы IV Всесоюзного семинара «Разработка и применение программных средств ПЭВМ в учебном процессе». Москва: ИПИ АН СССР, 1988.



В. КАЙМИН, Л. КУГЕЛЬ,  
Е. КУЗНИЦКИЙ, Н. УГРИНОВИЧ

## Алгоритмизация и начала программирования

13

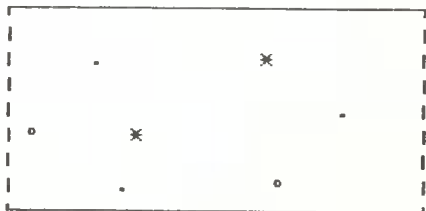
Одной из важнейших тем в изучении основ алгоритмизации является разбор примеров алгоритмов и программ с циклами. Рациональнее всего эту тему начинать с разбора циклов со счетчиком типа «от  $k=1$  до  $n$ », а уже после этого разбирать циклы с завершением по произвольному условию.

Наиболее ярким примером алгоритма с использованием цикла по счетчику являются алогоритм и программа вывода на экран ЭВМ «изображения звездного неба», разбор которых необходимо начинать с разбора сценария. В приводимом сценарии компьютер выводит на экран картинку «звездного неба» со случайным набором точек, число которых задается с клавиатуры ЭВМ:

Сценарий: "Звездное небо"

```
звезд = ? << n >>
```

↓  
v



Формирование картины «звездного неба» в соответствующих алгоритме и программе организуется с помощью цикла со счетчиком  $k$ :

Продолжение. Начало см.: Информатика и образование. 1989. № 2, 3, 4, 5.

алг звездное небо

```
нач
xm:=255:ym:=191
запрос ("звезд=",n)
- от k=1 до n цикл
: x:=случайное[0:xm]
: y:=случайное[0:ym]
: точка(x,y)
- кцикл
кон
```

```
10 REM звездное небо
20 XM=255:YM=191
30 INPUT "звезд=";N
[35 SCREEN 2]
40 FOR K=1 TO N
50 X=RND(1)*XM
60 Y=RND(1)*YM
70 PSET(X,Y),2
80 NEXT K
90 END
```

Общая схема таких циклов по счетчику и их реализация на Бейсике:

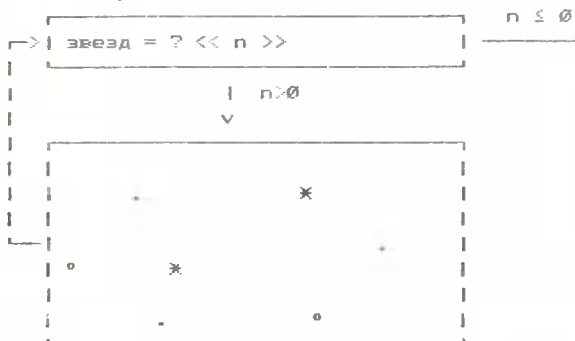
```
от k=1 до n цикл   100 FOR K=1 TO N
< действия >     ... <действия>
кцикл              200 NEXT K
```

где  $k$  — это некоторая целочисленная переменная, а  $l$  и  $n$  — это некоторые константы, переменные или выражения, задающие начальное и конечное значения для счетчика  $k$ , который будет последовательно принимать значения  $l$ ,  $l+1$ ,  $l+2$  и так далее вплоть до  $k=n$ .

Развитием этого примера может служить замена случайных точек на случайные отрезки, рамки или окружности и самостоятельное составление учениками соответствующих сценариев алгоритмов и программ, с которыми они будут проводить эксперименты на ЭВМ.

Следующий шаг в развитии этой темы — превращение данного алгоритма и программы в игру-эксперимент. Для этого нужно организовать цикл ввода «числа звезд» — значе- ний величины  $n$ . Сценарий и реализация этого эксперимента — алгоритм и программа получают вид:

Сценарий: "Звездное небо"



#### 14 Алгоритм:

```

алг "звездное небо"
нач
  xm=255:ym=191:cm=15
  запрос ("звезд=",n)
  пока n>=0 цикл
    от k = 1 до n цикл
      x:= случайное[0:xm]
      y:= случайное[0:ym]
      c:= случайное[0:cm]
      точка ( x,y ),c
    кцикл
  запрос ("звезд=",n)
кцикл
кон
  
```

Программа на Бейсике:

```

10 'звездное небо
20 xm=255:ym=191:cm=15
30 input "звезд=";n
40 if n <= 0 goto 140
45 screen 2
50 for k = 1 to n
60 x = rnd(1) * xm
70 y = rnd(1) * ym
80 c = rnd(1) * cm
90 pset(x,y),c
100 next k
110 input "звезд=";n
120 goto 40
130 'кцикл
140 end
  
```

Эксперименты с этой программой на ЭВМ следует заключить разбором общей схемы записи и кодирования на Бейсике циклоз с завершением по условию:

```

-->пока <условие> цикл ---
: <действия> :
!--кцикл <-----
-->40 IF NOT<условие> GOTO 100 -
: ... <действия> :
!--90 GOTO 40 <-----
100 'кцикл
  
```

Здесь важно разъяснить правила выполнения циклов такого вида. Их выполнение на каждом шаге состоит в проверке «условия», а затем — в выполнении действий, указанных между словами «цикл» и «кцикл». Повторение «действий» может продолжаться сколь угодно долго, пока выполняется «условие». Больше того, цикл вообще может продолжаться бесконечно, если это «условие» завершения цикла никогда не выполнится. Такие ситуации называются закливанием алгоритмов и программ. В то же время выполнение «действий» в цикле может вообще не начаться, если «условие» завершения окажется ложным на самом первом шаге выполнения цикла.

Тему циклов, равно как и всех основных средств организации алгоритмов и программ, разумно начинать с графических задач, отличающихся наглядным представлением результатов. Но далее необходимо показать и разъяснить их возможности для организации математических вычислений и обработки данных. Таким примером может служить организация диалоговых алгоритмов и программ суммирования последовательностей чисел и подсчета их среднего значения со сценарием такого вида:

Сценарий:

```

подсчет суммы:
чисел = << n >>
x(1) = << x1 >>
x(2) = << x2 >>
.....
x(n) = << xn >>
сумма = < Sn >
  
```

где  $S_n = x_1 + \dots + x_n$

```

алг подсчет суммы
нач
  вывод ("подсчет суммы")
  запрос ("чисел = ",n)
  s:=0
  от k=1 до n цикл
    вывод ("x(",k,")=")
    запрос (x)
    s:=s+x
  кцикл
  вывод ("сумма =",s)
кон
  
```

Особенность данного алгоритма состоит в том, что он предполагает ввод последовательности чисел  $x_1, x_2, \dots, x_n$  и подсчет общей суммы через подсчет промежуточных сумм  $s_1 = x_1, s_2 = x_1 + x_2, S_n = x_1 + x_2 + \dots + x_n$  с помощью повторного выполнения присваивания «s:=s+x» с каждой новой вводимой величиной  $x_k$ . Для того чтобы убедиться в этом, необходимо рассмотреть результаты выполнения операций, выполняемых в цикле и до него:



алг подсчет суммы	результаты:
нач	
...	
S:=0	{S0=0}
от k=1 до n цикл	
ввод (x)	{X=Xk}
s:=s+x	{Sk=Sk-1 + Xk}
цикл	{k=(1, ..., n)}
...	
кон	

рекуррентной формулой:

$$s_k = s_{k-1} \cdot \frac{k-1}{k} + x_k \cdot \frac{1}{k}$$

$$k = (1, \dots, n)$$

Правильность данного способа вычисления среднего можно показать с помощью схемы рассуждения, аналогичной рассмотренной выше. Реализацию этого алгоритма на языке программирования вполне можно доверить учащимся.

Сценарий:

«Обработка данных»

```

чисел = << n >>
x(1) = << x1 >> S = < S1 >
x(2) = << x2 >> S = < S2 >
.....
x(n) = << xn >> S = < Sn >
среднее = < Sn >

```

где  $S_n = (x_1 + \dots + x_n) / n$

алг обработка данных

```

нач
вывод ("обработка данных")
запрос ("чисел = ", n)
s:=0
от k=1 до n цикл
вывод ("x(", k, ")=")
запрос (x)
s:= s - (k-1)/k + x/k
вывод ("S = ", s)
кцикл
вывод ("среднее = ", s)
кон

```

Действительно, присваивание «s:=0» фиксирует первое начальное значение переменной s, равное  $S_0=0$ . Далее начинает выполняться цикл со счетчиком k, принимающим значения  $k=1, 2, \dots, n$ . На первом шаге при  $k=1$  вводится первое число  $x_1$  и переменной S присваиванием  $S:=S+x$  будет присвоено значение  $S_1=S_0+x_1=x_1$ . На втором шаге при  $k=2$  будет введено число  $x_2$  и присваиванием  $S:=S+x$  будет вычислено второе значение переменной S, которое станет равным  $S_2=S_1+x_2=x_1+x_2$ . Таким образом на втором шаге переменная S станет равной сумме первых двух членов последовательности  $x_1+x_2$ . На каждом последующем k-м шаге выполнения цикла будет соответственно вводиться очередное k-е число  $x_k$  и вычисляться очередная промежуточная сумма  $s_k$  и вычисляться очередная промежуточная сумма  $s_k=x_1+x_2+\dots+x_k$  первых k членов последовательности  $x_1, x_2, \dots, x_n$ .

В самом деле, на третьем шаге при  $k=3$  результатом будет  $s_3=s_2+x_3=x_1+x_2+x_3$ , при  $k=4$  будет  $s_4=x_1+x_2+x_3+x_4$ . Далее, полагая, что на  $k-1$ -м шаге результатом будет  $s_{k-1}=x_1+x_2+\dots+x_{k-1}$  из соотношения  $s_k=s_{k-1}+x_k$ , описывающего результат присваивания  $s:=s+x$  на k-м шаге, получаем  $s_k=x_1+x_2+\dots+x_{k-1}+x_k$ . И наконец, на последнем шаге при  $k=n$  будет введено последнее n-ое число  $x_n$  и значение переменной s станет равным  $s_n=x_1+x_2+\dots+x_n$ .

Использованные здесь равенства

$$s_k = s_{k-1} + x_k$$

$$k = (1, \dots, n)$$

являются простейшим примером рекуррентных соотношений, широко используемых в математике для различных схем вычислений. Реализация рекуррентных вычислений и является основным смыслом организации циклов в вычислительных алгоритмах.

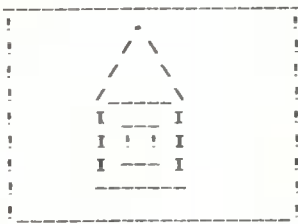
Развитием темы использования циклов для организации математических вычислений и обработки данных могут служить алгоритмы определения среднего, максимального или минимального значений числовых последовательностей. Например, для подсчета среднестатистического значения последовательности из чисел можно воспользоваться следующей

Составление алгоритмов определения максимумов и минимумов, а также анализ результатов их выполнения могут быть представлены ученикам для самостоятельной проработки. Главным здесь должна быть не только проверка на ЭВМ составленных алгоритмов и программ, но и самостоятельный разбор правильности составленных алгоритмов, который в принципе аналогичен приведенным выше рассуждениям.

Рассмотрение вспомогательных алгоритмов и использование в программах на Бейсике подпрограмм разумно также начинать с задач графического представления информации. Лучшей темой могут служить задачи построения на экране ЭВМ сложных изображений, состоящих из нескольких частей, например построение изображений «зданий», составляемых из «унифицированных блоков». Для этого следует вернуться к задаче построения изображения домика.

«Строительными блоками» здесь являются изображения крыши и стены с окном. Если в простейшем алгоритме построения изображения домика исходило из уточнения координат изображаемых частей, то теперь

Сценарий " Домик "



алг домик  
нач  
крыша  
стена с окном  
кон

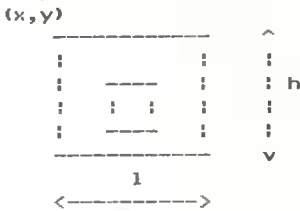
мы введем описания отдельных частей с выделением их геометрических размеров в качестве параметров, характеризующих эти строительные блоки.

Фрагмент "крыша"



алг крыша  
арг x,y: вещ  
l,h: вещ  
нач  
отрезок (x,y)-(x+l,y)  
отрезок (x,y)-(x+l/2,y-h)  
отрезок (x+l,y)-(x+l/2,y-h)  
кон

Фрагмент "стена с окном"



алг стена с окном  
арг x,y: вещ  
l,h: вещ  
нач  
рамка (x,y)-(x+l,y+h)  
рамка (x+l/3,y+h/3)-(x+l-2/3,y+h-2/3)  
кон

Изображение домика мы можем получить, указывая в алгоритме названия частей с указанием необходимых параметров, а в программе на Бейсике — записав вызовы соответствующих подпрограмм.

алг домик  
нач  
крыша (x=80,y=100,l=100,h=60)  
стена (x=80,y=100,l=100,h=80)  
кон

```
10 REM домик
20 SCREEN 2
30 X=80:Y=100
40 L=100:H=60
50 GOSUB 200 'вывод крыши
60 X=80:Y=100
70 L=100:H=80
80 GOSUB 400 'вывод стены
90 END
```

Отметим, что конкретные значения параметров подпрограмм на Бейсике приходится задавать присваиваниями. Описания подпрограмм на Бейсике, реализующих указанные вспомогательные алгоритмы, имеют вид:

```
200 'крыша
210 'X,Y - левый угол
220 'L,H - длина и высота
230 LINE (X,Y)-(X+L,Y),B
240 LINE (X,Y)-(X+L/2,Y-H),B
250 LINE (X+L,Y)-(X+L/2,Y-H),B
260 RETURN
400 'стена с окном
410 'X,Y - левый верхний угол
420 'L,H - длина и высота
430 LINE (X,Y)-(X+L,Y+H),3
440 LINE (X+L/3,Y+H/3)-(X+2/3*L,
y+2/3*H),3
450 RETURN
```

Ввод и испытание этих программ на ЭВМ рационально проводить «сверху вниз». Прежде всего нужно ввести подпрограмму вывода изображения крыши и соответствующий оператор вызова этой подпрограммы — 50 gosub 400. По завершении ввода этих операторов можно запустить программу на выполнение и посмотреть на экране получившееся изображение. После этого можно вводить вторую подпрограмму построения изображения стены с окном и оператор вызова этой подпрограммы — 80 gosub 400 и запустить на выполнение полную версию программы.

Использование этих подпрограмм позволяет строить на экране ЭВМ не только изображение такого простого домика, но и изображения более сложных двух- или трехсекционных домов, многоэтажных домов и даже поселков, указывая координаты расстановки соответствующих блоков. Например, для получения изображения двухсекционных или двухэтажных домов можно воспользоваться следующими алгоритмами,

"двухсекционный дом"



предоставляя учащимся самим записывать соответствующие программы на Бейсике и подбирать необходимые им координаты и параметры.

алг двухсекционный дом

нач

крыша (x=60, y=100, l=160, h=80)

стена (x=60, y=100, l=80, h=100)

стена (x=140, y=100, l=80, h=100)

кон

" двухэтажный дом "



алг двухэтажный дом

нач

крыша (x=100, y=60, l=80, h=40)

стена (x=100, y=60, l=80, h=60)

стена (x=100, y=120, l=80, h=60)

кон

Общая форма записи вспомогательных алгоритмов требует описания их аргументов и результатов:

алг название

арг <список аргументов>

рез <список результатов>

нач

<действия>

кон

400 'название

410 '<описание аргументов>

420 '<описание результатов>

...

... <действия>

500 RETURN

Аргументами и результатами вспомогательных алгоритмов и подпрограмм могут являться некоторые переменные или массивы, где должны размещаться исходные данные и результаты вычисления или обработки данных, выполняемых в этих алгоритмах и подпрограммах.

В Бейсике, к сожалению, нет средств явного описания параметров подпрограмм. Поэтому аргументы и результаты выполнения подпрограмм приходится описывать в форме комментариев (см. пример выше). В то же время в алгоритмах в целях большей наглядности указание конкретных значений аргументов становится более ясным при явном указании имен соответствующих параметров. В силу указанных причин запись обращений к вспомогательным алгоритмам и кодирование на Бейсике обраще-

ний к подпрограммам, реализующих вспомогательные алгоритмы, производится в такой форме:

уравнение (a=1, b=4, c=4; v=x1, x=x2)

60 a=1:b=4:c=4

70 GOSUB 800

80 V=X1:X=X2

где a, b, c — это аргументы, а x<sub>1</sub>, x<sub>2</sub> — результаты вспомогательного алгоритма, вычисляющего «корни уравнений». Для разделения аргументов и параметров в обращениях к вспомогательным алгоритмам можно указывать знак «точка с запятой».

Использование вспомогательных алгоритмов и подпрограмм для организации решения задачи на ЭВМ обычно связывается с решением некоторых вспомогательных задач или с выделением операций ввода-вывода данных в виде отдельных модулей. Главная часть такого рода модульных программ представляет общий план организации вычислений и обработки данных. Применение вспомогательных алгоритмов и подпрограмм в этих целях лучше всего продемонстрировать на решении геометрических задач с наглядным представлением результатов.

В качестве такого примера можно рассмотреть задачу определения двух наиболее удаленных точек среди n точек, заданных на плоскости случайным образом. Для проведения экспериментов на ЭВМ с этой задачей рекомендуется взять сценарий, в котором на экране ЭВМ изображается n точек и должен быть построен отрезок, соединяющий пару точек, наиболее удаленных среди них.

точек = << n >>

:



алг поиск наиболее удаленных точек

арг x[1:n], y[1:n]:вещ

нач

запрос ("точек=", n)

генерация точек

поиск пары точек

вывод искомого отрезка

кон

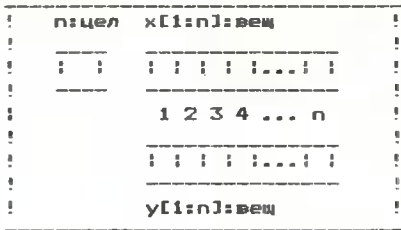
В предлагаемом алгоритме и программе для хранения координат точек используются два массива x[1:n] и y[1:n], которые можно представить рисунком «распределения памяти в ЭВМ»:

OldPC.su

7002

музей компьютеров

память ЭВМ



Приведенный рисунок продолжает основную методическую идею представления переменных и массивов в виде картинок с «распределением» памяти ЭВМ. В такой трактовке вполне естественны объяснения и переменных, и массивов в алгоритмах и программах как некоторых областей машинной памяти, в которых могут храниться и обновляться данные. И если переменные служат для размещения в памяти отдельных величин — чисел, слов и т. п., то массивы используются для размещения совокупностей однотипных данных. При этом для указания отдельных элементов массивов используются индексы — один, два, три или более. Количество индексов определяется описаниями массивов, которые задаются обычно в начале алгоритмов и программ и в которых указываются максимальные и минимальные значения этих индексов.

В программах на Бейсике описания массивов задаются с помощью операторов dim — операторов указания размеров массивов. В рассматриваемой программе массивы x [1:n] и y [1:n] с координатами точек можно описать, например, оператором dim x(100), y(100), который задает описание двух массивов x[1:100] и y[1:100] объемом по 100 вещественных чисел.

Приведем полный алгоритм и программу на Бейсике поиска наиболее удаленной пары точек, соответствующие выбранному сценарию, в которых используются указанные массивы.

```

алг поиск удаленных точек
нач x[1:100], y[1:100]:вещ
запрос ("точек=", n)
генерация точек (n)
выбор пары точек (im, jm)
линия (x(im), y(im)) - (x(jm), y(jm))
кон
    
```

```

10 'поиск удаленных точек
20 DIM X(100), Y(100)
30 INPUT "точек="; N
40 GOSUB 200 'генерация точек
50 GOSUB 400 'выбор пары точек
60 LINE (X(IM), Y(IM)) - (X(JM), Y(JM)), B
70 END
    
```

```

алг генерация точек
нач x:=255; y:=191
от k=1 до n цикл
x[k]:=случайное[0:x]
y[k]:=случайное[0:y]
точка (x[k], y[k])
кцикл
кон
    
```

```

200 'генерация точек
210 XM=255; YM=191
[215 SCREEN 2]
220 FOR K=1 TO N
230 X(K)=RND(1)*XM
240 Y(K)=RND(1)*YM
250 PSET (X(K), Y(K)), 2
260 NEXT K
270 RETURN
    
```

```

алг выбор пары точек
рез im, jm:цел
нач
gm:=0; im:=1, jm:=1
от i=1 до n-1 цикл
от j=i+1 до n цикл
r:=расстояние (i, j)
если r<gm то
gm:=r; im:=i; jm:=j
[линия(x[i], y[i]) - (x[j], y[j])]
кесли
кцикл
кон
    
```

```

400 'выбор пары точек
410 'IM, JM - номера точек
420 RM:=0; IM:=1; JM:=1
430 FOR I=1 TO N-1
440 FOR J=I+1 TO N
450 GOSUB 600 'r=расстояние(i, j)
460 IF R<RM GOTO 490
470 RM=R; IM=I; JM=J
[480 LINE (X(I), Y(I)) - (X(J), Y(J)), 2]
490 'кесли
500 NEXT J
510 NEXT I
520 RETURN
    
```

```

алг расстояние между точками
арг i, j: цел
рез r:вещ
нач
dx:=x[i]-x[j]
dy:=y[i]-y[j]
r:=J(dx^2+dy^2)
кон
    
```

```

600 'расстояние между точками
610 'I, J - индексы точек
620 'R - расстояние между точками
630 DX=X(I)-X(J)
640 DY=Y(I)-Y(J)
650 R=SQR(DX^2+DY^2)
660 RETURN
    
```

Последний вспомогательный алгоритм выделен в целях иллюстрации: как в алгоритмах и программах, может выделяться вычисление сложных функций. Проведем

18

анализ результатов выполнения присваиваний в этом алгоритме:

Присваивания:	Результаты:
$dx := x[i] - x[j]$	$\{ dx = x[i] - x[j]$
$dy := y[i] - y[j]$	$\{ dy = y[i] - y[j]$
$r := \sqrt{dx^2 + dy^2}$	$\{ r = \sqrt{dx^2 + dy^2}$

Подставив выражения для  $dx$  и  $dy$  в последнюю формулу, получим формулу, описывающую конечный результат выполнения этого вспомогательного алгоритма:

$$r = \sqrt{((x[i] - y[j])^2 + (y[i] - y[j])^2)}$$

Данная формула есть не что иное, как определение расстояния между двумя точками на плоскости.

Ввод и отладку данной программы можно и нужно проводить по частям. Первой должна быть введена и исполнена подпрограмма поиска, которая проводит перебор точек, заменив в ней вычисление расстояния на вывод отрезка, соединяющего соответствующие точки. Убедившись в полноте перебора всех пар точек вывод отрезка, следует заменить на вычисление расстояния между точками с отладочным выводом выделяемых пар точек. По завершении отладки программы отладочный вывод промежуточных пар точек можно удалить. В результате будет получено окончательное решение задачи на ЭВМ.

После получения решения можно провести анализ правильности алгоритма в целом. Выделим с этой целью его наиболее существенную часть — алгоритм перебора пар точек и выбора среди них наиболее удаленных.

присваивание  $gm := 0$  фиксирует начальное значение переменной  $gm$ , которое становится равным 0. Далее два вложенных цикла — цикл по  $i$  от 1 до  $n-1$  и цикл по  $j$  от  $i+1$  до  $n$  организуют перебор пар индексов с номерами — (1,2), (1,3), ..., (1,n) для  $i=1$ , затем пар (2,3), (2,4), ... (2,n) для значения  $i=2$  и так далее вплоть до значения  $i=n-1$ , при котором внутренний цикл обеспечит обработку пары точек с номерами  $(n, n+1)$ .

Внутри циклов для каждой пары индексов  $(i, j)$  вычисляется величина  $r_{ij}$  — расстоянию  $(i, j)$ , которая далее сравнивается со значением переменной  $gm$ . Если величина  $r_{ij}$  оказывается больше текущего значения переменной  $gm$ :  $r_{ij} > gm$ , то значение  $r_{ij}$  присваивается переменной  $gm$  и соответствующие индексы  $(i, j)$  фиксируются в переменных  $im$  и  $jm$ . Вполне понятно, что текущие значения  $gm, im, jm$  переменной  $gm$  есть не что иное, как наибольшие среди расстояний между точками:

$$gm_{ij} = \max(r_{12}, r_{13}, \dots, r_{ij}).$$

Остается объяснить, почему такой способ перебора пар точек исчерпывает все варианты. Действительно, поскольку расстояния между точками  $r_{ij}$  и  $r_{ji}$  равны друг другу  $r_{ij} = r_{ji}$ , а  $r_{ii} = 0$ , то перебор пар точек с индексами  $(i, j)$ , такими, что  $i < j$ , обеспечит полноту перебора для поиска наибольшего расстояния среди всех пар точек.

Тему организации обработки данных с помощью массивов в алгоритмах и программах можно развить на задаче поиска перевода по словарю. Пусть у нас имеется некоторый двуязычный словарь, например словарь из русских и английских слов.

```

алг выбор пары точек
нач
  gm := 0
  im := 1; jm := 1
  от i=1 до n-1 цикл
    от j=i+1 до n цикл
      r := расстояние(i, j)
      если r > gm то
        gm := r
        im := i; jm := j
      кесли
    кцикл
  кцикл
кон

```

результаты :

```

gm = 0
(i, j) = [(1, 2), (1, 3), ..., (1, n),
          (2, 3), (2, 4), ..., (2, n),
          ..., (n-1, n)]
r / i j = расстояние(i, j)
gm / i j = { r / i j, при r > gm / i j
            { gm / i j, в ином случае

```

Рассмотрим результаты выполнения операций, составляющих этот алгоритм. Первое

#### СЛОВАРЬ:

-----	:
русск	: англ
-----	:
печать	: print
запрос	: input
точка	: pset
линия	: line
если	: if
то	: then
...	: ...
конец	: end
-----	:

```

алг перевод по словарю
нач
  загрузка словаря
  запрос ("слово =", слв)
  k := 1
  пока k <= n и русск[k] <= слв цикл
    k := k + 1
  кцикл
  если слв = русск[k] то
    вывод("перевод =", англ[k])

```

```

иначе
вывод ("перевода - нет")
кесли
кон

```

Особенность данного алгоритма в том, что он организует перебор слов в словаре, который размещается в массивах: русск[1:n] и англ[1:n]. Построение цикла таково, что его действие состоит всего лишь в проверке совпадения введенного слова и k-го слова в словаре русск[1:n] и в проверке — исчерпан или нет словарь: k=n? При невыполнении одного из этих условий цикл завершается. Далее организуется вывод перевода найденного слова по словарю английских слов англ[1:k] с перепроверкой слова в русском словаре.

Реализация данного алгоритма на Бейсике требует не только его кодирования, но решения проблемы организации данных — словарей русских и английских слов. Для их размещения и обработки потребуются два массива rs[1:n] и an[1:n], которые в программе будут описаны оператором:

```
20 DIM RS$(100), AN$(100)
```

Для задания словаря в программе можно воспользоваться операторами data, позволяющими описывать различные последовательности данных. Пример описания:

```

1000 DATA печать, print
1010 DATA запрос, input
...
2000 DATA *,*

```

Считывание данных из таких описаний проводится с помощью операторов read, например оператор read s1\$, s2\$ (чтение (s1, s2)). Здесь s1\$, s2\$ — переменные, значениями которых, по правилам языка Бейсик, являются не числа, а строки из символов. Повторение выполнения этого оператора data приведет к считыванию следующей пары слов и т. д. В качестве признака конца слов мы воспользуемся парой символов — \*, \*. Считывание этих символов будет означать конец данных.

Алгоритм и программа на Бейсике поиска перевода слов по словарю, задаваемому в программе операторами data, получает такой вид:

```

алг перевод по словарю
нач русск[1:n], англ[1:n]:симв
загрузка словаря
запрос ("слово =", слв)
к:=1
пока к<=n и русск[k]≠слв цикл
к:=к+1
кцикл
если слв=русск[k] то
вывод ("перевод =", англ[k])

```

```

инес слв$русск[k] то
вывод ("перевода нет")
кесли
кон

```

```

10 REM перевод по словарю
20 DIM RS$(100), AN$(100)
30 GOSUB 200 'загрузка словаря
40 INPUT "слово ="; SL$
50 K=1
60 IF RS$(K)=SL$ OR K=N GOTO 90
70 K=K+1
80 GOTO 60
90 'кцикл
100 IF SL$=RS$(K) THEN
PRINT "перевод =" ; AN$(K)
110 IF SL$<>RS$(K) THEN
PRINT "перевода нет"
120 END

```

алг загрузка словаря  
нач

```

n:=0
чтение (s1, s2)
пока s1#"*" и s2#"*" цикл
n:=n+1
русск[n]:=s1
англ[n]:=s2
чтение (s1, s2)
кцикл

```

кон

```

200 'загрузка словаря
210 'N - число слов
220 N=0
230 READ S1$, S2$
240 IF S1$="*" OR S2$="*" GOTO 300
250 N=N+1
260 RS$(N)=S1$
270 AN$(N)=S2$
280 READ S1$, S2$
290 GOTO 240
300 'кцикл
310 RETURN

```

дан" словарь "

нач

```

-----
Iпечать print I
Iзапрос I input I
Iточка I pset I
Iлиния I line I
Iесли I if I
Iто I then I
Iконец I end I
I... I ... I

```

кон

```

I_____I

```

```

1000 'словарь
1010 data печать, print
1020 data запрос, input
1030 data точка, pset
1040 data линия, line
1050 data если, if
1060 data то, then
1070 data конец, end
2000 data *,*

```



```

600 'вычисление: Y=f(X)
610 'X - аргумент
620 'Y - значение
630 IF X>=0 THEN Y=X
640 IF X<0 THEN Y=-X
650 RETURN

```

Ввод этой программы рационально проводить по частям. Первой ввести и проверить подпрограмму ввода данных, затем — подпрограмму построения графика, заменив в ней вызов подпрограммы вычисления значений функции — операторы 460 gosub 600 на оператор 460  $y=x*x$ . После проверки этой версии программы можно ввести и подключить последнюю часть — подпрограмму вычисления значений функции.

При организации работ на ЭВМ в целях экономии времени подпрограммы ввода исходных данных и построения графика могут быть заготовлены заранее и рассылаться по сети. Ученикам в таком случае следует дать задание на самостоятельное составление подпрограмм вычисления значений функций. Примеры такого рода функций: и т. п.

```

y = e\X\,
y = sin(x),
y = e\X\ - sin(x),
y = e\X\ - cos(x),
y = (x-1) - x - (x+1),
y = sin(x) - sin(10-x)

```

В таких заданиях полезной является постановка задач исследования функций на различных интервалах и в различных диапазонах. Целью исследований могло бы быть нахождение особых точек: максимумов, минимумов, точек перегиба и т. п.

Одной из последних при изучении начал программирования для ЭВМ должна быть тема составления диалоговых программ. Создание диалоговых программ проще всего проводить и изучать на основе сценарно-алгоритмического подхода, допускающего разделение труда в составлении сценариев, алгоритмов и программ. Последней подтемой в освоении этого подхода является создание диалоговых программ, в которых управление работой ЭВМ проводится посредством команд, что наиболее характерно для персональных ЭВМ.

Конкретной иллюстрацией таких средств организации диалога может служить программа вычерчивания линий на экране ЭВМ с использованием клавиш-команд на клавиатуре. Базовым средством для реализации подобного рода программ на персональных ЭВМ являются операторы опроса клавиш, нажатых на клавиатуре. В языке Бейсик роль такого средства играет оператор `inkey$`, указывающий символ, надписанный на нажатой клавише. Просмотр на экране всех символов, имеющихся в ПЭВМ, можно

организовать с помощью такого рода простой программы:

```

алг набор символов ЭВМ
нач
от k=0 до 255 цикл
вывод (симв(k))
кцикл
кон
10 REM набор символов ЭВМ
20 FOR K=0 TO 255
30 PRINT CHR$(K)
40 NEXT K
50 END

```

Здесь функция `chr$(k)` сообщает символ, кодируемый на ЭВМ числом  $k$ , называемый ASC-кодом. Таблицы кодировки символов в ASC-коде можно найти в справочнике по языку программирования либо увидеть на экране ЭВМ с помощью такой диалоговой программы:

```

алг коды клавиш
нач
вывод ("коды клавиш")
пока c#"s" цикл
с:=клавиша
если c#" " то
вывод ("код(",c,""),c)
кесли
кцикл
кон
10 REM коды клавиш
20 PRINT "коды клавиш"
30 IF C#"s" GOTO 70
40 C#=INKEY$
50 IF C#>" " THEN PRINT ASC(C#),C
60 GOTO 30
70 'кцикл
80 END

```

Обратите внимание: в этом алгоритме конец работы программы связан с нажатием клавиши «s» (stop). При нажатии любой другой клавиши на экране ЭВМ будет появляться соответствующий символ и его машинный код.

После освоения всех перечисленных средств организации диалога можно дать полное определение понятия сценария для диалоговых программ. Сценарии — это: 1) описания всех текстов и рисунков, появляющихся на экране ЭВМ в ходе диалога; 2) описания правил ввода данных и управление работой ЭВМ; 3) описания всех реакций ЭВМ на вводимые данные и команды; 4) описание логики перехода от одной сцены (картинки, кадра) к другим сценам в зависимости от вводимых и накопленных данных.

Примером диалоговой программы с использованием клавиш-команд и немедленной



реакцией машины на эти команды могут служить алгоритм и программа вычерчивания линий на экране ЭВМ. Предлагаемая программа работает в соответствии со следующим сценарием:

Сценарий " Черчение на ЭВМ "

```

I-----I
I-----I
I      I . I      I
I      I__I      I
I-----I
I-----I
I-----I
I-----I
I-----I

```

Команды черчения:

н - вниз      п - вправо  
л - влево    s - стоп  
в - вверх

Работа на ЭВМ с данной программой состоит в нажатии клавиш «л», «п», «н», «в». Нажатие клавиши «н» приведет к прочерчиванию линии вниз, клавиши «л» — влево, клавиши «в» — вверх, а клавиши «п» — вправо. Нажимая на эти клавиши, можно нарисовать на экране достаточно затейливые узоры. Для завершения работы достаточно нажать клавишу «s» — стоп.

алг черчение на ЭВМ

```

нач
  xm:=255,ym:=191
  x:=xm/2;y:=ym/2
  c:=""
  пока с#="s" цикл
    точка (x,y)
    c:=клавиша
    если с="л" то
      x:=x-1
    инес с="п" то
      x:=x+1
    инес с="в" то
      y:=y-1
    инес с="н" то
      y:=y+1
    кесли
  кцикл
кон
10 REM черчение на ЭВМ
[15 SCREEN 2]
20 XM=255;YM=191
30 X=XM/2;Y=YM/2
40 C$=""
50 IF C$="s" GOTO 130
60 PSET (x,y),2
70 C$=INKEY$
80 IF C$="л" THEN X=X-1
90 IF C$="п" THEN X=X+1
100 IF C$="в" THEN Y=Y-1
110 IF C$="н" THEN Y=Y+1
120 GOTO 50
130 *кцикл
200 END

```

Развитие этой программы возможно в двух направлениях. Первое — ввести новые команды рисования типа «o» — окружность и «p» — рамка с рисованием окружностей и рамок, «ц» — смена цвета и т. п. Второе направление — введение динамики, изменив роль нажатия клавиш так, чтобы нажатие клавиш значило бы не шаг сдвига, а импульс ускорения. Смещение точки (x, y) при этом должно пересчитываться по формулам:

$$\begin{cases} x' = x + dx & dx = \begin{cases} -1 & \text{при } c = \text{"л"} \\ +1 & \text{при } c = \text{"п"} \end{cases} \\ y' = y + dy & dy = \begin{cases} -1 & \text{при } c = \text{"в"} \\ +1 & \text{при } c = \text{"н"} \end{cases} \end{cases}$$

Соответствующий диалоговый алгоритм и моделирующая программа могут быть такими:

алг модель движения

```

нач
  xm:=255;ym:=191
  x:=xm/2;y:=ym/2
  c:="" ; dx:=0,dy:=0
  пока с#="s" цикл
    точка (x,y)
    c:=клавиша
    если с="л" то
      dx:=-1
    инес с="п" то
      dx:=+1
    инес с="в" то
      dy:=-1
    инес с="н" то
      dy:=+1
    кесли
    x:=x+dx
    y:=y+dy
  кцикл
кон
10 REM модель движения
[15 SCREEN 2]
20 XM=255;YM=191
30 X=XM/2;Y=YM/2
40 C$="" ; DX=0;DY=0
50 IF C$="s" THEN 160
60 PSET (X,Y)
70 C$=INKEY$
80 IF C$="л" THEN DX=-1
90 IF C$="п" THEN DX=+1
100 IF C$="в" THEN DY=-1
110 IF C$="н" THEN DY=+1
120 *кесли
130 X=X+DX
140 Y=Y+DY
150 GOTO 50
160 *кцикл
170 END

```

Работа с этой программой, как правило, вызывает значительное оживление у учеников, поскольку это уже некоторая игра. Машина начинает сама вычерчивать некоторую траекторию движения, а мы можем воздействовать на это движение, нажимая

клавиши «в» — вверх, «н» — вниз, «п» — вправо, «л» — влево. Развитием этой моделирующей программы может быть введение несколько усложненной модели движения, более приближенной к управляемому движению материальных тел:

$$\begin{cases} x' = x + dx & ax = \begin{cases} -1 & \text{при } c = "л" \\ +1 & \text{при } c = "п" \end{cases} \\ dx' = dx + ax \end{cases}$$

$$\begin{cases} y' = y + dy & ay = \begin{cases} +1 & \text{при } c = "н" \\ -1 & \text{при } c = "в" \end{cases} \\ dy' = dy + ay \end{cases}$$

Реализация в алгоритме и программе этой модели движения, близкой к реальным законам движения тел под воздействием сил, полностью превращается в игру с машиной. Особенностью здесь является возможность быстрого разгона движущейся точки, которая скоро может выйти за пределы экрана. Суть игры может состоять и в удержании точки в определенных пределах либо в обходе некоторого набора точек, заданных на экране машиной. При оформлении этих точек в виде разноцветных звезд и планет эта игра может быть подана как один из вариантов космических путешествий.

Многообразие дальнейших заданий и задач для составления диалоговых алгоритмов и программ весьма велико. Учитель может и должен сам подбирать задачи для самостоятельного составления программ учащимися в зависимости от уровня их подготовленности и интересов.

Завершением данной части курса должна быть зачетная работа учащихся с самостоятельным составлением алгоритмов и программ по сценарию, предложенному или согласованному с учителем. Привлекательной особенностью сценарно-алгоритмического подхода к составлению диалоговых программ является возможность коллективной организации работ учащихся. При таком подходе можно проводить совместные обсуждения сценариев, организацию взаимопомощи и взаимоконтроля при руководстве со стороны учителя.

Принципиальным в рамках сценарно-алгоритмического подхода становится совсем новое понимание программирования как «искусства написания программ». Акценты

«искусства» в основном смещаются в сторону составления сценариев программ. И именно здесь активно обсуждаются вопросы наглядности и эстетичности рисунков, ясности и понятности текстов, дружелюбности и тактичности сообщений, занимательности сюжета и т. п. Собственно составление алгоритмов и программ, исходя из предложенных сценариев, становится для учащихся ясным, понятным и легко управляемым видом деятельности.

Итогом работы учащихся в изучении начал программирования и основ алгоритмизации должно стать формирование и закрепление следующих знаний, навыков и умений:

- 1) знания основных вычислительных возможностей ЭВМ;
- 2) умения составлять простейшие алгоритмы и программы для ЭВМ;
- 3) умения проводить отладку программ на персональных ЭВМ;
- 4) знания возможностей простейшего языка программирования;
- 5) знания правил описания и составления алгоритмов;
- 6) знания правил описания и составления сценариев для диалоговых программ.

*Продолжение следует.*

## Л и т е р а т у р а

1. *Каймин В. А., Питеркин В. М.* Основы информатики и вычислительной техники: Учебное пособие для учителей. М.: Изд. МИЭМ, 1985.
2. Основы информатики и вычислительной техники: Пробное учебное пособие для 9—10 классов средних школ / Под редакцией А. П. Ершова, В. М. Монахова. М.: Просвещение, 1985.
3. *Каймин В. А., Щеголев А. Г., Ерохина Е. А., Федюшин Д. П.* Основы информатики и вычислительной техники: Пробный учебник для 9—10 классов средних школ. М.: Просвещение, 1989.
4. *Каймин В. А., Угринович Н. Д.* О преподавании курса основ информатики и вычислительной техники по машинному варианту // Информатика и образование. 1989. № 2.
5. *Угринович Н. Д.* Информатика: теория, методика, задачи. М.: Изд. МГИУУ, 1988.



А. КУШНИРЕНКО

## О новом учебнике

### 9. Команда повторения (цикл) пока

Использование в алгоритме только команд-приказов приводит к выполнению строго определенной, заранее известной последовательности действий — ЭВМ лишь приказывает исполнить какие-то действия, но никак не анализирует их результаты.

Представьте себе начальника, который отдает приказы, но не получает никаких донесений о результатах их исполнения; шофера, ведущего автомобиль с закрытыми глазами. Понятно, что так далеко не уедешь.

Для составления алгоритмов решения сложных задач, например задач управления Роботом, надо иметь «обратную связь» — уметь получать от Робота информацию об обстановке, в которой он оказался. Это можно сделать с помощью команд-вопросов: «клетка закрашена?», «на севере свободно?» и др. При исполнении команды-вопроса между ЭВМ и Роботом происходит диалог. ЭВМ задает Роботу вопрос, Робот анализирует обстановку и отвечает «да» или «нет».

Использование команд-вопросов позволяет составлять циклические и разветвляющиеся алгоритмы, работающие в разных ситуациях по-разному. Именно в этом и состоит основное отличие Робота от Черепашки в языке Лого.

Подчеркнем еще раз, что конструкции алгоритмического языка вводятся «по необходимости», для решения наглядных задач по управлению Роботом. Например, для введения цикла пока в учебнике рассматривается такая задача: известно, что где-то западнее Робота есть стена; расстояние от Робота до стены неизвестно; нужно подвести Робота вплотную к стене.

Если мы командуем Роботом «вручную», без ЭВМ, то можно поступить так. Спросим сначала: «на западе свободно?». Если Робот ответит «нет», значит, он уже у стены — задача решена и больше ничего делать не надо. Если же Робот ответит «да», то он еще не у стены, значит, можно скомандовать

«шаг на запад» и опять спросить: «на западе свободно?». Если Робот ответит «нет», значит, он у стены, а если ответит «да», значит, не у стены. В последнем случае надо опять командовать «шаг на запад», опять спрашивать «на западе свободно?» и т. д.

Какая именно последовательность команд Робота будет при этом исполнена (скомандуем ли мы Робота «шаг на запад» 0, 3, 8, 1990 или иное число раз), заранее сказать нельзя — это зависит от обстановки (от начального расположения Робота относительно стен на поле).

Далее в учебнике рассматривается задача составления алгоритма, при исполнении которого ЭВМ автоматически, без участия человека, переведет Робота в требуемое положение. При этом естественно возникает необходимость в записи приведенной выше последовательности действий на алгоритмическом языке, что и служит основой для введения цикла пока:

алг на запад до стены

нач

· нц пока на западе свободно

· · шаг на запад

· кц

кон

После освоения циклов и раз и для форма записи новой команды не вызывает затруднений. Смысл же новой команды может быть пояснен на примере.

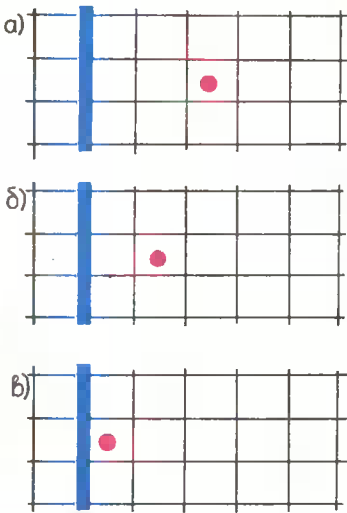
Пусть в начальный момент Робот находится в следующей обстановке (рис. 9 а, б, в):

Тогда при исполнении алгоритма «на запад до стены» диалог ЭВМ — Робот будет таким: ЭВМ: на западе свободно? Робот: да  
ЭВМ: шаг на запад Робот: смещается на запад:

ЭВМ: на западе свободно? Робот: да  
ЭВМ: шаг на запад Робот: смещается на запад:

ЭВМ: на западе свободно? Робот: нет

Поскольку Робот ответил «нет», т. е. записанное после пока условие не соблюдается, то на этом исполнение цикла пока и алгоритма «на запад до стены» заканчивается.



ложение Робота в этом прямоугольнике заранее не известны. Один и тот же алгоритм должен переводить Робота в нужное положение во всех случаях.

25. Известно, что Робот находится внутри прямоугольника, огороженного со всех сторон стенами. Робот стоит где-то у западной стены. Составьте алгоритм, после выполнения которого Робот закрасит горизонтальную полосу клеток от исходного положения Робота до восточной стены, а затем вернется в исходное положение.

26. На поле Робота стен нет, но есть прямоугольник из закрашенных клеток, причем Робот находится в юго-западном углу этого прямоугольника. Составьте алгоритм, переводящий Робота в северо-восточный угол прямоугольника.

27. Какие команды ЭВМ будет давать Роботу при исполнении команды повторения:

а) нц пока клетка **не** закрашена  
· закрасить клетку

б) нц пока клетка закрашена  
· закрасить клетку

кц  
в ситуации, когда Робот стоит: 1) в закрашенной клетке, 2) в незакрашенной?

28. Дано, что на поле Робота одна стена, идущая строго с запада на восток. Робот находится в одной из клеток, прилегающих к стене с севера. Точные размеры стены и точное расположение Робота неизвестны. Составьте алгоритм, при исполнении которого Робот:

а) переместится в одну из клеток, прилегающих к стене с юга;

б) закрасит все клетки, прилегающие к стене с севера;

в) закрасит все клетки, прилегающие к стене с юга;

г) закрасит все прилегающие к стене клетки (и с севера, и с юга).

29. Дано, что Робот находится внутри прямоугольника, огороженного с четырех сторон стенами. Внутри прямоугольника стен нет. Составьте алгоритм, при исполнении которого Робот закрасивает все клетки внутри прямоугольника, прилегающие к стенам.

30. Составьте алгоритм «выход из радиоактивной зоны на север (арг вещ  $a$ )», аналогичный алгоритму «вход в радиоактивную зону» (Робот вне радиоактивной зоны, если уровень радиации в клетке не превосходит  $a$ ).

31. Дано, что Робот находится в северо-западном углу в прямоугольнике, огороженном с четырех сторон стенами. Внутри прямоугольника имеется горизонтальная стена с одним проходом, идущая от западного до восточного края прямоугольника (проход не прилегает ни к западной, ни к восточной стене прямоугольника). Составьте алгоритм,

26 Использование «диалогов» ЭВМ — Робот (в учебнике и несколько) для иллюстрации работы ЭВМ в процессе выполнения цикла пока и наглядное графическое представление результатов каждого шага позволяют объяснить смысл конструкции пока, ее свойства и особенности на простых, понятных примерах.

Команды-вопросы, однако, лишь частный случай команд «обратной связи», т. е. команд, в результате исполнения которых автоматическое устройство сообщает ЭВМ какую-то информацию. В учебнике рассматриваются еще две такие команды:

вещ уровень радиации

вещ температура

Служебное слово вещ означает, что в результате исполнения этих команд ЭВМ получает одно вещественное число — уровень радиации или температуру соответственно. С использованием этих команд можно решать задачи, связь которых с жизнью очевидна: алг вход в радиоактивную зону (арг вещ  $a$ )

дано !на поле Робота стен нет, южнее  
!Робота расположена

!радиоактивная зона

надо !Робот сместился на юг в первую  
!клетку, уровень

!радиации в которой превышает  $a$

нач

· нц пока уровень радиации  $\leq a$

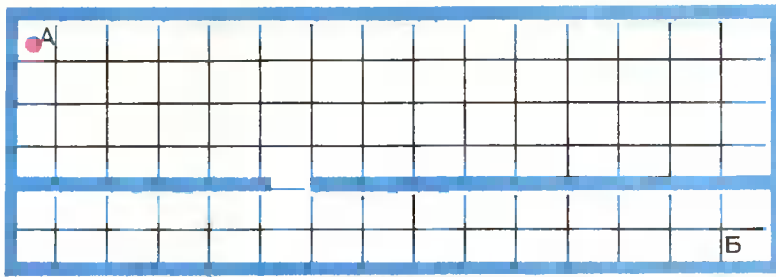
· · шаг на юг

· кц

кон

### Задачи

24. Известно, что Робот находится внутри прямоугольника, огороженного со всех сторон стенами. Внутри прямоугольника стен нет. Составьте алгоритм, переводящий Робота в северо-западный угол этого прямоугольника. Ни размеры прямоугольника, ни распо-

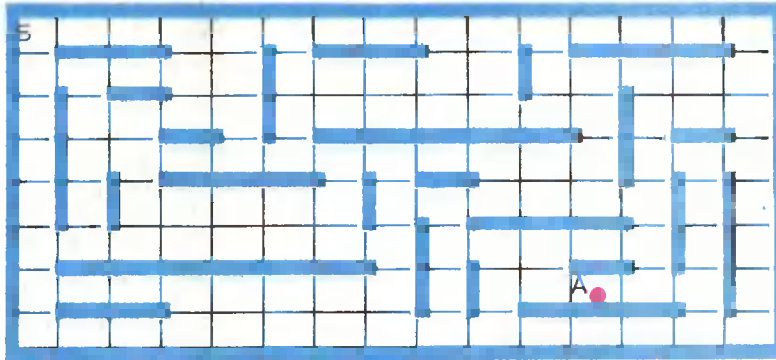


при исполнении которого Робот переместится в юго-восточный угол прямоугольника (рис. 10).

32. Дано, что Робот находится в какой-то клетке внутри лабиринта, имеющего вид прямоугольника и огороженного с четырех сторон стенами. Стены внутри лабиринта имеют вид отрезков прямых и не касаются друг друга и наружных стен прямоугольника (рис. 11). Составьте алгоритм, при исполнении которого Робот переместится в северо-западный угол любого такого лабиринта.

Указание: можно использовать условия вида «на севере свободно или на западе свободно».

11



### 10. Введение условий, команд ветвления и контроля

Введение в учебнике команд-вопросов привело к расширению понятия условие. Теперь в условия могут входить как команды-вопросы («на юге свободно?»), так и отношения между величинами (« $a \geq b$ »). Результатом проверки любого условия является либо да (если условие соблюдается), либо нет (если условие не соблюдается). Составное условие можно составить из простых, пользуясь операциями и, или, не. Операцию не разрешается для удобства чтения расположить «внутри» команды-вопроса: вместо не «на севере свободно» разрешается написать «на севере не свободно» и т. п.

В учебнике сохранены команды ветвления если и выбор и введена команда контроля утв (от слова утверждение), позволяющая

ЭВМ контролировать выполнение предусмотренных человеком условий в процессе исполнения алгоритма.

Разумеется, все эти команды вводятся на наглядном материале. Например, для комплекса Робот-измеритель в учебнике рассмотрена такая задача: Робот стоит в левом конце горизонтального коридора. Требуется разметить (закрасить) те клетки коридора, в которых уровень радиации превышает предельно допустимое значение  $a$ .

алг разметка опасных клеток (арг вещь  $a$ )  
дано !Робот стоит в левой (западной) клетке горизонтального коридора  
дано !Робот вышел из коридора на восток,

!закрашены клетки коридора, в которых уровень радиации выше  $a$

нач

- нц пока на юге не свободно
- · если уровень радиации  $> a$
- · · то закрасить
- · все
- · шаг на восток

· кц

кон

*Задачи*

33. Запишите на алгоритмическом языке условие, которому удовлетворяют точки, изображенные на рис. 12.

34. Составьте алгоритм со следующим заголовком:

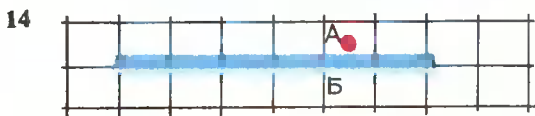
а) алг закрасить тупики  
дано !Робот стоит в левой (западной) клетке горизонтального коридо-



личин, авторы считают, что при работе в старших классах более раннее введение понятия величины и команды присваивания вряд ли может чему-то повредить: величины с успехом можно проходить параллельно с аргументами вспомогательных алгоритмов и правилами линейной записи выражений.

### Введение величин

Использование исполнителей позволяет не только долгое время обходиться без величин, но сделать процесс введения величины плавным и естественным, комбинируя в алгоритме изменение величины и положения Робота. Рассмотрим, например, следующую задачу: Робот расположен в клетке А севернее стены неизвестной (но конечной) протяженности, идущей с запада на восток. Надо перевести Робота «сквозь стену» — в клетку Б (рис. 14):



При непосредственном (ручном) управлении Роботом мы бы могли, например, двигать его на восток до края стены и по дороге считать число сделанных шагов, после чего сместить Робота на юг и двигать его обратно на запад — на запомненное число шагов.

Для составления соответствующей программы надо уметь приказывать ЭВМ запоминать, менять и использовать информацию о числе сделанных шагов, что естественно приводит к введению целочисленной величины и командам работы с ней.

С использованием величин решение задачи записывается так:

алг на юг сквозь стену

дано !южнее Робота горизонтальная стена, других стен нет

надо !Робот обошел стену и стоит на клетку южнее исходного положения

нач цел  $n$  ! число шагов, сделанных Роботом на восток

- $n := 0$
- нц пока на юге не свободно
- шаг на восток
- $n := n + 1$
- кц

!Робот миновал восточный край стены

- шаг на юг
- нц  $n$  раз
- шаг на запад
- кц

кон

Как и в учебном пособии, в учебнике вводятся имя, значение и тип величины. Чтобы школьники легче освоились с этими не-

простыми понятиями, в учебнике дается следующая наглядная модель: память ЭВМ представляется в виде обычной классной доски, на которой можно записывать информацию, читать ее, стирать, записывать заново и т. д.

Место, отводимое в памяти ЭВМ для запоминания отдельной величины, представляется в виде прямоугольника, над которым написаны тип и имя этой величины:

цел  $n$



Внутри прямоугольника записывается значение величины. Если в прямоугольнике ничего не написано, то это значит, что величина не определена.

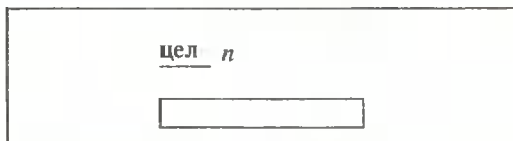
ЭВМ отводит величине место в памяти следующим образом: в начале исполнения алгоритма ЭВМ выделяет для него часть памяти. Эта часть памяти также изображается прямоугольником на доске, а выше него пишется название алгоритма:

алг на юг сквозь стену.



Встретив после слова нач описание «цел  $n$ », ЭВМ отводит для запоминания величины  $n$  часть памяти внутри прямоугольника алгоритма:

алг на юг сквозь стену

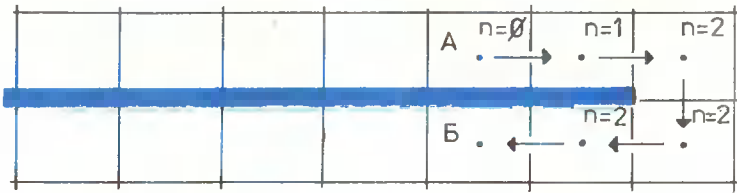


При исполнении команды присваивания « $n := 0$ » ЭВМ записывает значение выражения (т. е. 0) внутри прямоугольника величины « $n$ » (старое значение при этом будет затираться).

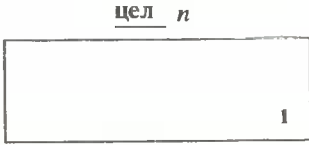
цел  $n$



Таким образом, после исполнения команды « $n := 0$ » величина  $n$  будет иметь значение 0. Если теперь исполнить команду « $n := n + 1$ », то ЭВМ сначала вычислит значение выражения « $n + 1$ », т. е. заменит имя величины  $n$  на ее значение (0), и вычислит  $0 + 1 = 1$ . После этого ЭВМ сотрет старое значение внутри прямоугольника величины  $n$  и запишет на это



место полученное новое значение (1):



В целом движение Робота и изменения величины  $n$  в процессе исполнения алгоритма можно описать одним рисунком (в клетках указаны значения величины  $n$  в момент выхода Робота из клетки) (см. рис. 15).

30 Много наглядных алгоритмов с использованием величин можно придумать для комплекса Робот-измеритель. Вот пример:

**алг** закрасить радиоактивные клетки  
**дано** !Робот стоит западнее коридора, уходящего на восток  
**надо** !закрашены все клетки коридора, в которых уровень радиации выше, чем !в исходном положении Робота.  
 !Робот вышел из коридора на восток  
**нач вещь**  $x!$  уровень радиации в исходной клетке

.  $x := \text{уровень радиации}$   
 . шаг на восток  
 . **нц** пока на юге не свободно  
 . . **если** уровень радиации  $> x$   
 . . . **то** закрасить  
 . . . **все**  
 . шаг на восток  
 . **кц**

**кон**

**Задачи**

36. Значение величины  $x$  равно 3. Чему оно будет равно после исполнения команды:

а)  $x := 5$ ; б)  $x := x + 5$ ; в)  $y := x$ ?

37. После исполнения команды:

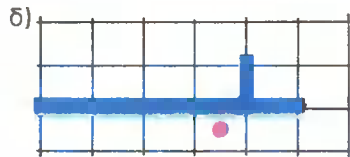
а)  $x := x + 5$ ; б)  $x := -x$ ; в)  $y := x$ ;

или серии команд:

г)  $y := 1$  д)  $y := x$

$x := x + y$   $x := y$

значение величины  $x$  стало равно 3. Чему было равно значение величины  $x$  до исполнения?



38. После исполнения команды присваивания « $x := x + y$ » значение величины  $x$  равно 3, а значение у равно 5. Чему были равны значения величин  $x$  и  $y$  до исполнения команды?

39. Значение величины  $x$  равно  $a$ , значение у равно  $b$ . После исполнения какой из указанных ниже серий команд значения величин  $x$  и  $y$  поменяются, т. е.  $x$  станет равно  $b$ , а  $y$  станет равно  $a$ :

а)  $x := y$  б)  $t := x$   
 $y := x$   $x := y$   
 $y := t$

в)  $x := x + y$  г)  $t := x$   
 $y := x - y$   $y := t$   
 $x := x - y$   $x := y$

40. Опишите действия ЭВМ и Робота при исполнении алгоритма «на юг сквозь стену» для начальных положений Робота, указанных на рис. 16.

41. Составьте алгоритм со следующим заголовком:

а) **алг** на юг до стены закрасить и вернуться

**дано** !где-то южнее Робота есть стена  
**надо** !Робот дошел до ближайшей стены на юг, закрасил !клетку и вернулся в исходное !положение

б) **алг** закрасить клетку в северо-восточном углу

**дано** !Робот где-то внутри прямоугольника, огороженного стенами, других стен нет

**надо** !Робот закрасил клетку в северо-восточном углу прямоугольника !и вернулся в исходное положение

в) **алг** отойти вдвое дальше от южной стены

**дано** !где-то южнее Робота есть стена, ! других стен нет

**надо** !Робот севернее стены на расстоянии вдвое большем, чем !исходное



г) алг симметрия

дано !где-то южнее Робота есть стена  
!длиной в одну клетку, других  
!стен нет  
надо !Робот оказался в положении,  
!симметричном исходному отно-  
!сительно стены

42. Робот находится в северо-западном (левом верхнем) углу прямоугольника, огороженного стенами. Составьте алгоритм, после исполнения которого Робот будет стоять в клетке с минимальным уровнем радиации.

43. Робот находится западнее коридора, идущего на восток. Составьте алгоритм, после исполнения которого Чертежник нарисует график уровня радиации в этом тупике.

44. Робот находится в северо-западном (левом верхнем) углу прямоугольника, огороженного стенами. Внутри прямоугольника есть стены, идущие с запада на восток. Ни одна из них не примыкает к западному краю прямоугольника. Составьте алгоритм, после исполнения которого Чертежник нарисует расположение стен на поле Робота внутри прямоугольника.

## 12. Введение табличных величин (массивов)

Как и для других элементов языка программирования (циклов, ветвлений, величин и т. п.), для мотивации введения табличных величин в учебнике используются «внешние» задачи, формулировка которых с таблицами никак не связана. Пусть, например, Робот стоит в западной клетке 100-клеточного горизонтального коридора, а требуется вывести его из коридора на восток и вычислить количество «опасных» клеток коридора, уровень радиации в которых выше, чем в клетке выхода. (Радиация в этой клетке принята за предельно допустимую).

Одно из решений выглядит так: сначала проведем Робота по коридору с запада на восток, выведем его в клетку выхода и запоем уровень радиации в этой клетке. Потом проведем Робота обратно и по дороге подсчитаем число клеток с повышенной радиацией. Наконец, проведем Робота по коридору еще раз, чтобы вывести его из коридора на восток, как требуется в условии задачи. В итоге Робот трижды пройдет по опасному коридору.

Использование табличных величин позволяет отделить получение информации от ее обработки и составить программу, при исполнении которой Робот пройдет по коридору всего один раз — с запада на восток, по дороге запоминая в элементах таблицы уровни радиации в клетках коридора. Достигнув клетки выхода, заполненную информа-

цию можно обработать и подсчитать требуемое число опасных клеток:

алг число клеток с повышенной радиацией  
(рез цел  $n$ )

дано !Робот стоит в левой (западной)  
!клетке 100-клеточного коридора,  
!уходящего на восток

надо !Робот вышел из коридора на восток,  
!вычислено количество клеток коридора,  
!уровень радиации в которых выше,  
!чем в клетке выхода (т. е. !в первой  
!клетке вне коридора на !восток)

нач вещ  $a0$ , цел  $i$ , вещ таб  $a[1:100]$

. нц для  $i$  от 1 до 100

. .  $a[i]$  := уровень радиации

. . шаг на восток

. кц

. утв  $!a$  := значения уровня радиации в  
!клетках коридора

!Робот в клетке выхода

.  $a0$  := уровень радиации ! в клетке выхода

.  $n$  := 0

. нц для  $i$  от 1 до 100

. . если  $a[i] > a0$

. . . то  $n := n + 1$

. . все

. кц

. утв  $!n$  := число клеток с повышенной радиацией

кон

Секрет могущества ЭВМ — высокая скорость работы, большая память, возможность быстро перерабатывать громадные объемы информации. Введение табличных величин знакомит учащихся со способами компактной записи больших объемов информации. В приложении к учебнику рассматриваются задачи кодирования различных видов встречающейся в нашей жизни информации с помощью табличных величин: кодирование графической информации, кодирование информации о проданных железнодорожных билетах и др.

Отметим, что при использовании Робота многие популярные в программировании задачи по работе с таблицами: нахождение максимального элемента массива, подсчет числа положительных элементов массива, определение индекса максимального элемента, вычисление среднего арифметического элементов массива и др. — получают наглядную геометрическую интерпретацию: нахождение максимального уровня радиации в коридоре, нахождение числа радиоактивных клеток коридора и др. Различие между линейными и прямоугольными (одномерными и двумерными) массивами, понятия элемента таблицы и его индекса также становят-

ся геометрически наглядными при описании обстановки на поле Робота с помощью прямоугольных таблиц.

45. Известно, что значением целочисленной табличной величины  $k$  является четверка чисел (3, 1, -2, 4). Чему равно: а)  $k[1]$ ; б)  $k[3]$ ; в)  $k[5]$ ; г)  $k[k[1]]$ ?

46. Известно, что значением целочисленной табличной величины  $k$  является четверка чисел (3, 1, -2, 4). Вычислите значение величины  $k$  после исполнения серии команд:

а)  $k[1]:=k[4]$ ;  $k[1]:=k[1]+k[2]$

б)  $i:=3$ ;  $k[1]:=k[i]\times k[i+1]$

Задана линейная таблица  $k[1:4]$ . Запишите выражение, значением которого является:

а) удвоенное значение третьего элемента таблицы;

б) сумма первого и второго элементов таблицы;

в) произведение всех элементов таблицы.

47. Составьте алгоритмы со следующими заголовками:

а) алг обмен (арг цел  $n$ , арг рез вещь таб  $a[1:n]$ )

надо !первый и последний элементы таблицы поменялись местами

б) алг перестановка (арг цел  $n$ , арг рез вещь таб  $a[1:n]$ )

надо !минимальный элемент таблицы переставлен в ее начало, а первый — на место минимального

в) алг вещь максимум (арг цел  $n$ , вещ таб  $a[1:n]$ )

надо !знач=значение максимального элемента в таблице  $a$

г) алг цел индекс максимума (арг цел  $n$ , вещ таб  $a[1:n]$ )

надо !знач=индекс максимального элемента в таблице  $a$

д) алг цел число положительных (арг цел  $n$ , вещ таб  $a[1:n]$ )

надо !знач=число положительных элементов в таблице  $a$

е) алг цел число больших (арг цел  $n$ , вещ таб  $a[1:n]$ , вещ  $a0$ )

надо !знач=число элементов таблицы  $a$ , больших  $a0$

ж) алг копия (арг цел  $n$ , вещ таб  $a[1:n]$ , рез вещь таб  $b[1:n]$ )

надо !элементы таблицы  $a$  скопированы в таблицу  $b$  в том же порядке

з) алг копия (арг цел  $n$ , вещ таб  $a[1:n]$ , рез вещь таб  $b[1:n]$ )

надо !элементы таблицы  $a$  скопированы в таблицу  $b$  в обратном порядке:  $b[1]=a[n]$ ,  $b[2]=a[n-1]$ , ...

и) алг корректировка (арг цел  $n$ , арг рез цел таб  $a[1:n]$ )

надо !все элементы отчета  $a$ , меньшие 100, заменены на 100

к) алг заполнение нулями (арг цел  $n$ , рез цел таб  $a[1:n]$ )

надо !все элементы  $a$  равны нулю

48. Составьте алгоритмы со следующими заголовками:

а) алг цел произведение (арг цел  $n$ , арг цел таб  $a[1:n]$ )

надо !знач=произведение элементов таблицы  $a$

б) алг вещь среднеарифметическое (арг цел  $n$ , вещ таб  $a[1:n]$ )

надо !вычислено среднее арифметическое элементов  $a$

в) алг уровни радиации (арг цел  $m$ , рез вещь таб  $a[1:m, 1:n]$ )

дано !Робот в северо-западном углу прямоугольника размером  $m\times n$  !клеток, внутри стен нет

надо !в таблицу  $a$  размером  $m\times n$  занесены уровни радиации соответствующих клеток прямоугольника, !Робот в исходном положении

г) алг график (арг вещь  $a$ , арг вещь  $b$ , арг вещь таб  $y[1:n]$ )

дано  $a < b$  и  $n \geq 2$  !в таблице  $a$  заданы  $n$  значений !функции на отрезке  $[a, b]$

надо !Чертежник построил приближенный график функции

д) алг ломаная (арг цел  $n$ , вещ таб  $x[1:n]$ , вещ таб  $y[1:n]$ )

дано  $n \geq 2$  в таблицах  $x$  и  $y$  заданы координаты вершин ломаной линии

надо !Чертежник нарисовал эту ломаную линию

е) алг простые числа (арг цел  $n$ , рез цел таб  $p[1:n]$ )

дано  $n \geq 1$

надо !в таблице  $p$  — первые  $n$  простых чисел

ж) алг упорядочение (арг цел  $n$ , арг рез цел таб  $a[1:n]$ )

надо !элементы  $a$  упорядочены по неубыванию, т. е. переставлены !так, что  $a[1] \leq a[2] \leq \dots \leq a[n]$

### 13. Введение символьных и литерных величин

В отличие от пробных учебных пособий в новом учебнике сначала вводятся символьные (сим) величины. Значением такой величины может быть один символ: русская или латинская заглавная или строчная буква, цифра, знак препинания или специальный знак (например, знаки арифметических операций «+», «-», «×», «/», знаки отношений «<», «=» и др.).

Всего символьная величина может принимать 256 различных значений, занумерованных от 0 до 255.

Линейные таблицы, элементами которых являются символы, в алгоритмическом языке называются литерными величинами и описываются служебным словом лит.

Значением литерной величины является последовательность символов переменной длины. Такую последовательность часто называют строкой. Для обозначения *i*-го элемента литерной величины *t* в учебнике используется запись *t[i]*. В учебных пособиях требовалось писать *t[i:i]*. В новом учебнике такая запись тоже допустима, хотя она и менее удобна.

В алгоритме литерные значения записываются в виде последовательности символов, заключенных в кавычки. Например, можно написать:

$t := \text{«информатика»}$

После этой команды  $t[1] = \text{«и»}$ ,  $t[2] = \text{«н»}$ ,  $t[3] = \text{«ф»}$ , ...,  $t[11] = \text{«а»}$ :

лит *t*

и	н	ф	о	р	м	а	т	и	к	а
1	2	3	4	5	6	7	8	9	10	11

#### Задачи

49. Строка *t* равна «апельсин». Чему равны строки:

- $t[2:3] + t[8] + t[5]$ ;
- $t[6] + t[2] + t[11] + t[8] + t[7] + t[3:5]$ ;

50. Строка *t* равна «вертикаль». По образцу упр. 49 составьте из нее слова:

- «ветка»;
- «кирка»;
- «кильватер».

51. Составьте алгоритм со следующим заголовком:

а) алг цел количество предложений (арг лит *x*)

дано Строка *x* состоит из нескольких предложений,

!каждое из которых кончается точкой, восклицательным или вопросительным знаком

надо !знач=количество предложений !в строке *x*

б) алг обмен (арг сим *x*, *y*, арг рез лит *a*)

надо !всюду в «а» *x* заменен на *y*, а *y* заменен на *x*

в) алг лит перевертыш (арг лит *a*)

надо !знач= «перевертыш», если !строка «а» совпадает с собой !после переворачивания, и !знач=«не перевертыш» в противном случае

г) алг замена (арг лит *x*, *y*, арг рез лит *a*)

дано длин (*x*)=длин (*y*)

надо !всюду в строке «а» *x* заменено на *y*

### 14. Команды ввода-вывода информации 33 в алгоритмическом языке

Справедливые нарекания вызывало отсутствие в пробных учебных пособиях команд ввода-вывода информации. В учебнике этот недостаток исправлен. Вводятся эти команды как средства организации совместной работы человека и ЭВМ, удобные при решении многих задач. Для организации совместной работы необходимо предусмотреть в алгоритме обмен информации между человеком и ЭВМ. Информация при этом выводится на экран и вводится с клавиатуры.

В алгоритмическом языке предусмотрены три специальные команды ввода-вывода: ввод, вывод, нс. Команда нс (новая строка) указывает, что вывод информации должен быть продолжен на новой строке. Вот два примера использования команд ввода-вывода:

алг произведение

нач цел *m*, *n*

• вывод «Ведите два числа:»

• ввод *m*, *n*

• вывод нс, «Произведение чисел равно»,

*m*×*n*

кон

алг квадраты

нач цел *m*, *k*

• вывод «Введите количество квадратов»:

• ввод *k*

• нц для *m* от 1 до *k*

• • вывод нс, *m*×*k*2

• кц

кон

#### Задачи

52. Составьте алгоритм, который:

а) запрашивает у человека пять целых чисел и выводит квадраты этих чисел;

б) запрашивает у человека два числа и выдает их полусумму;

в) запрашивает у человека два натуральных числа — числитель и знаменатель

дроби — и выдает результат сокращения этой дроби;

г) \* запрашивает у человека натуральное число и выдает разложение этого числа на простые множители.

*Продолжение следует.*

**В. ДОЛМАТОВ**  
НИИ СиМО АПН СССР

## Методические проблемы построения курса информатики для V—VI классов

34

Сегодня все отчетливее осознается необходимость переноса курса информатики из старших классов в средние. На это указывают как все чаще появляющиеся публикации по этой проблеме, так и реальные попытки учителей-экспериментаторов, организуемых кружки и факультативы по информатике для семи-, пяти- и даже четвероклассников.

Большинству учителей и методистов ясно, что информатика как учебный предмет в старших классах «опаздывает» с формированием навыков использования компьютера, что многие формируемые в рамках этого предмета навыки являются не узкопредметными, а общеобразовательными. «Опускание» курса ОИВТ в средние классы позволило бы:

использовать приобретенные учащимися знания и умения на других учебных предметах;

активнее развивать познавательные способности учащихся, используя компьютер в качестве инструмента решения учебных задач;

формировать у учащихся основы научного мировоззрения за счет более раннего, чем сейчас, привития им навыков работы с моделями явлений на компьютере.

При этом надо отчетливо понимать, что перенос курса потребует не просто адаптации (как поступают большинство исследователей, работающих в этой области) содержания предмета к возрастным особенностям и возможностям школьников, а кардинального пересмотра как содержания, так и методики преподавания.

Проблема заключается в необходимости дать в руки учителя методические средства, обеспечивающие решение задач, стоящих перед таким курсом. Без них все благие намерения введения информатики в этой

группе классов (обучение основам компьютерной грамотности, развитие познавательных способностей, формирование информационной культуры и т. д.) останутся декларацией, не имеющей отношения к реальным итогам учебного процесса.

Целью статьи является попытка, обобщив зарубежный и отечественный опыт преподавания информатики (в том числе и собственный), а также современные тенденции в понимании места этого предмета в системе общего среднего образования, сформировать целостный взгляд на цели, содержание и методику преподавания курса информатики для V—VI классов.

Следует подчеркнуть, что в разработанных у нас и за рубежом программах цели обучения имеют заметный уклон в сторону освоения навыков программирования. На наш взгляд, такое видение содержания информатики как общеобразовательного предмета изживет себя, уступив место введению в основы алгоритмизации, овладению навыками грамотной постановки и формализации задач. Это позволит раскрыть весь спектр связей ОИВТ с другими учебными предметами.

При формулировке целей обучения мы исходили из сегодняшнего понимания задач курса информатики, отраженного в работах А. П. Ершова, Е. Б. Машбица, А. А. Кузнецова, В. А. Каймина и других. Эта группа целей в значительной степени совпадает со сформулированными в программе курса ОИВТ для IX—X классов. Отличие состоит в постановке двух дополнительных целей (отражающих наше понимание места информатики в общем среднем образовании):

формирование навыков структурного анализа задачи,

формирование навыков формализации предметных задач.

Проведенный нами анализ концепций курса, разрабатываемых разными авторскими коллективами (в Москве, Новосибирске, Переславле-Залесском, Свердловске), показал, что многие из них делают упор на понятие «исполнитель алгоритма». Но приходится констатировать, что, во-первых, понятие «исполнитель» рассматривается ими в отрыве от понятия «задача», а во-вторых, сам выбор исполнителей осуществляется произвольно. В основном семейство исполнителей представлено такими исполнителями, как: Чертежник, Робот, Инженер (Москва, Свердловск), Раскрашка, Буквоед (Переславль-Залесский).

Мы считаем, что с методической точки зрения выбор исполнителей должен быть обоснован, и они должны образовывать систему взаимодополняющих друг друга исполнителей. Такая система специализированных исполнителей была нами выделена. При этом каждый специализированный исполнитель представляет собой как бы модель прикладной программы.

С другой стороны, отношение «задача — исполнитель» рассматривалось нами как генетически исходное для формирования основ компьютерной грамотности (рис. 1). Работа с указанным отношением при постановке учебных задач позволяет не только продемонстрировать учащимся происхождение понятий, составляющих теоретический фундамент компьютерной грамотности (действие, команда, данные, результат, алгоритм, программа и др.), но и подойти в процессе знакомства с системой исполнителей к понятию универсального исполнителя, к числу ко-

торых может быть отнесен и сам компьютер.

Но это — логическое видение содержания изучаемого материала. Для его методического воплощения необходим некоторый наглядный «образ», на основе которого можно организовать эффективное усвоение этой логики. Наглядность играет особую роль при организации обучения в V—VI классах, и выбору ее формы нами было уделено особое внимание.

В качестве учебной модели («образа») универсального исполнителя мы выбрали систему исполнителей, каждый из которых реализует ограниченный набор действий, отвечающих разным предметным областям, т. е. является специализированным.

Выбор состава исполнителей, включаемых в систему, определяется кругом решаемых предметных задач и возможностью его расширения. В качестве базисных в состав системы были включены следующие программируемые исполнители:

1. **Кодировщик** — моделирует процесс кодирования и декодирования различных видов информации (числовой, буквенной, графической, звуковой), а также перевод чисел из одной системы счисления в другую.

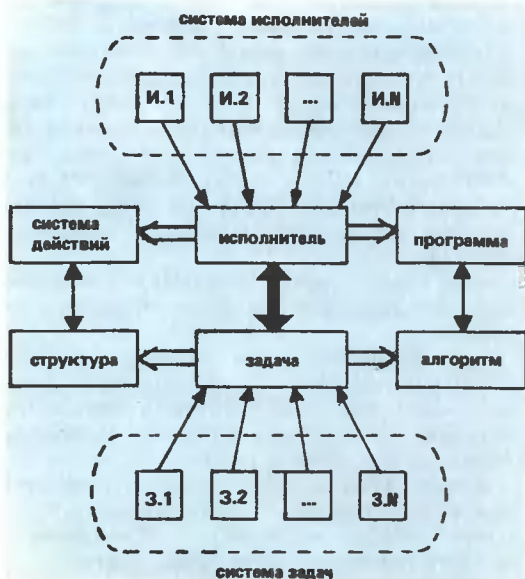
2. **Хранитель** — моделирует процесс хранения информации в той или иной форме (массив, очередь, список) и ее выдачу по запросу. Позволяет работать со структурой данных (образовать массив, присвоить или изменить имя, включить новый элемент, исключить элемент и т. д.), работает с объектами типа: число, буква, слово, фигура.

3. **Сортировщик** — реализует систему действий, связанную с перегруппировкой хранимой информации (сравнить элементы, обменять местами, найти элемент по признаку и т. д.), работает с объектами типа: число, буква, слово, фигура.

4. **Вычислитель** — реализует систему арифметических действий (модель калькулятора), работает с числовыми величинами.

5. **График** — обеспечивает построение простейших графических моделей (отвечающих объему программы по математике для этой группы классов) физических, химических, биологических явлений, формируя графическое представление информации в форме таблиц, графиков, диаграмм, работает с парами чисел (в прямоугольной системе координат).

6. **Логик** — реализует систему логических действий типа «И», «ИЛИ», «НЕ» и обеспечивает демонстрацию механизма организации работы сложных алгоритмов, включающих ветвление, цикл, выбор (например, отгадывание числа или буквы на основе постановки группы альтернативных вопросов и т. д.).



7. **Корректор** — моделирует действия человека в процессе правки текста с минимальным набором действий: исправление, вставка, стирание, перенос в конце строки.

8. **Робот** — моделирует набор типичных производственных операций: перемещение грузов, обработка деталей (придание им нужной формы, сверление и т. д.), сборка, покраска, контроль качества.

9. **Собеседник** — реализует систему действий, связанную с вводом и выводом информации на начальном (конечном) этапе исполнения алгоритма, с выводом диагностирующих сообщений или запросов на недостающие данные, фактически может обеспечивать интерактивный режим работы сконструированного исполнителя, что позволяет моделировать достаточно широкий круг предметных задач.

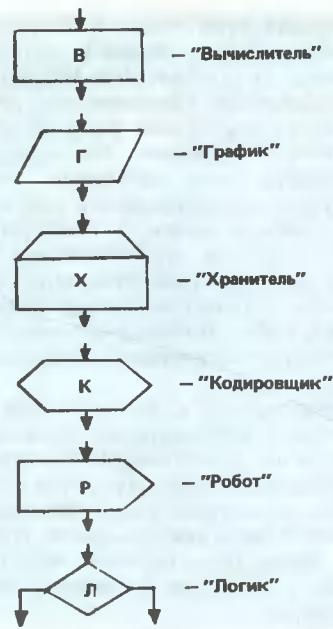
10. **Диспетчер** — осуществляет управление процессом вызова специализированных исполнителей, передачу параметров, анализ промежуточных результатов (в форме сравнения с эталонами) и выбор дальнейшего порядка работы коллектива исполнителей.

Для расширения круга привлекаемых предметных задач имеет смысл с самого начала заложить (на программном уровне) возможность расширения группы исполнителей.

Выбор имен-названий произволен и связан со стремлением сформировать у школьника достаточно наглядный (на основе аналогии с названиями профессий) образ подлежащей формализации деятельности. При этом процесс формализации как бы сливается с процессом конструирования «коллективного исполнителя», что создает благоприятные для усвоения учебного материала психологические условия. Использование условных знаков позволяет компактно описать взаимодействия специализированных исполнителей, входящих в коллектив с помощью языка блок-схем. Система является открытой и допускает расширение в рамках единого подхода, опирающегося на идею описания процесса решения любой задачи (учебной или производственной) через взаимодействие специализированных исполнителей.

На каждого исполнителя составляется «паспорт», являющийся его описанием, и ему ставится в соответствие знак — в виде некоторого геометрического символа (рис. 2).

Но ключевым свойством исполнителей, входящих в систему, придающим ей свойство методического инструмента обучения, является не специализация (которая лишь снимает избыточность и дублирование), а способность к кооперации и взаимодействию. Средством, обеспечивающим такое взаимодействие, оказывается информационная сеть, объединяющая специализированных испол-



нителей в «коллектив». Именно взаимодействие и обмен результатами работы придают системе исполнителей принципиальное новое свойство — способность реализовывать любую сколь угодно сложный алгоритм.

Использование предлагаемой модели (коллектив исполнителей) позволяет учебную задачу, взятую из той или иной предметной области, как бы «раскладывать» в системе базисных «линейно независимых» исполнителей. Работа с учебной системой исполнителей позволяет не только формировать навыки общения (диалога) с формальным исполнителем (в форме вызова и работы с прикладной программой), но и выработать навык структурного анализа, относящегося к числу важнейших и пока еще слабо формируемых общеобразовательных навыков той или иной учебной задачи. При этом сам структурный анализ задачи оказывается тесно связанным с процессом его формализации и алгоритмического описания, поскольку:

выделение структуры задачи отвечает отбору исполнителей, способных обеспечить ее решение;

формализация задачи связана с отображением некоторых второстепенных деталей для того, чтобы иметь возможность передать отдельные ее части для обработки формальному исполнителю;

алгоритмизация проявляется в необходимости спланировать такую последовательность работы отобранных исполнителей, которая обеспечит достижение цели.

Именно это позволяет рассматривать введенное нами понятие «система исполнителей» как методическое средство для достижения описанных выше целей обучения.

Воплощение предлагаемого подхода в рамках экспериментального курса требует поэтапного решения четырех задач:

1. Формирование знаний об основных принципах управления исполнителем (в процессе работы с готовыми исполнителями) и способах планирования его действий, т. е. привитие элементов алгоритмической культуры.

2. Формирование понятия о системе исполнителей (являющейся моделью пакета прикладных программ) как инструменте решения широкого класса предметных задач.

3. Выработка навыков формализации задач, взятых из различных предметов, и их структурного описания на основе понятия «коллектив исполнителей».

4. Формирование понятия о компьютере как об уникальном (перепрограммируемом) исполнителе.

Это и определило логику построения курса и распределение учебного материала по полугодиям.

Организация обучения в рамках описанного методического подхода предполагает разработку основных средств, образующих учебно-методический комплекс (УМК) по предмету и включающих:

- учебное пособие (для учащихся);
- систему заданий теоретического и практического характера для коллективной и индивидуальной форм учебной работы;
- пакета ППС, обеспечивающего программную поддержку.

На протяжении трех лет в ряде школ г. Москвы проводился эксперимент по преподаванию информатики в V—VI классах, в ходе которого проверялась эффективность предлагаемого подхода. Кроме того, проверялась методическая эффективность использования в обучении различных языковых средств (Бейсика, Пролога, ЛОГО, алгоритмического языка) и специальных учебных программных средств (типа «система специализированных исполнителей»).

Как показал эксперимент, использование того или иного языка программирования в качестве учебного оказывает существенное влияние на содержание и направленность задач, предъявляемых учащимся, структуру системы формируемых понятий и используемые организационные формы проведения занятий: Опыт показал, что обоснованное сочетание коллективных и индивидуальных форм организации учебной работы школьников, при ведущей роли коллективной формы, способно обеспечить лучшее усвоение изучаемого ма-

териала, развивает самостоятельность и активность учащихся.

В итоге нами был сделан вывод о тесной взаимосвязи используемых программных средств с другими компонентами методической системы обучения. Предложенная нами система программных средств — «коллективно-распределенная учебная среда» — оказалась адекватной сформулированным выше целям и выбранному содержанию курса, позволила воплотить в методике идею о ведущей роли коллективных форм работы и оказалась эффективным инструментом решения методических проблем, связанных с переносом курса информатики в V—VI классы.

В заключение приведем макет программы курса для V класса (без перечня практических работ и требований к знаниям и умениям в силу ограниченности объема статьи).

Автор был бы признателен всем, кто захочет высказаться (устно — по телефону 403-32-96 или письменно) по затронутой в статье проблеме или присоединиться к эксперименту.

37

## Программа

### I полугодие

1. Введение. Компьютер как техническое устройство (8 ч)

Технические устройства — исполнители человеческих действий. Круг решаемых задач.

Знакомство с компьютером как исполнителем: сбор, хранение и выдача информации, управление другими исполнителями (станком, роботом, технологическим процессом), рисование, исполнение музыкальных программ, игровой автомат.

Знакомство с устройством компьютера: общий вид, правила включения и выключения, техника безопасности при работе на ЭВМ, назначение отдельных блоков (клавиатура, дисплей, дисковод, принтер).

Понятие об информационной сети (ИС). Обмен информацией между компьютерами в ИС. Возможность распределения функций между компьютерами в ходе решения сложных задач. Коллектив исполнителей.

2. Задача — исполнитель — алгоритм (26 ч)

Задача и исполнитель. Исполнитель, типы исполнителей: формальный, неформальный, пошаговый, автомат. Система действий и система команд, «паспорт» исполнителя. Примеры различных исполнителей-автоматов. Характерные черты работы исполнителей-автоматов.

Планирование работы исполнителя. Алго-

ритм — план работы исполнителя. Зависимость алгоритма от системы действий исполнителя.

Специализированный исполнитель и его свойства. Система исполнителей. Исполнение и контроль, команды диагностирующие сообщения. Обратная связь. Диалог с исполнителем.

Коллектив исполнителей и решение сложных задач. Коллектив исполнителей — модель задачи. Информационное взаимодействие исполнителей и его описание (на языке блок-схем).

Информационная сеть — канал обмена информацией в системе исполнителей. Понятность, полнота и достоверность передаваемой информации. Основные принципы взаимодействия исполнителей в информационной сети. Перспективы развития информационных и вычислительных сетей.

### II полугодие

3. Алгоритм — программа — исполнитель (18 ч)

Что значит — решить задачу? Метод последовательного уточнения.

Алгоритм и его свойства. Библиотека алгоритмов. Структура алгоритма — отражение структуры задачи. Типы структур: линейная, циклическая, с выбором.

Языки описания алгоритмов (блок-схема, алгоритмический язык). Сходство и различие алгоритмического языка и языка естественного. Можно ли «разговаривать» с компьютером на человеческом языке?

Алгоритм и программа. Правила записи алгоритмов и программ. Программа как

управляющая информация. Программа — модель исполнителя.

4. Исполнитель и переработка информации (8 ч)

Промежуточный результат и возможность организации сложного поведения исполнителя. Логический и игровой автоматы, алгоритм их работы. Организация исполнения сложных алгоритмов с ветвлением и циклом (исполнитель Логик и его использование).

Кодирование сообщений с помощью нулей и единиц. Системы счисления и разные способы кодирования. Двоичная форма представления информации: чисел, букв, звуков, изображений, команд (исполнитель Кодировщик и работа с различными видами информации).

5. Универсальный исполнитель (8 ч)

Знакомство с системой базисных исполнителей: Редактор — текстовый и графический, Вычислитель, Хранитель, Логик, Кодировщик, Робот и др.); ее свойства: полнота, универсальность и т. д.

Понятие о сложном исполнителе как коллективе простых (на примере электронных таблиц, учебной компьютерной игры).

Зависимость свойств коллектива исполнителей от свойств входящих в него исполнителей и характера их взаимодействия. Моделирование сложного исполнителя на системе простых. «Диспетчер» как координатор взаимодействия исполнителей.

Понятие о коллективном исполнителе как исполнителе универсальном, т. е. способном решать различные задачи.

6. Экскурсия (2 ч)

38

## Персональная супер-ЭВМ

Сотрудники лаборатории архитектур высокопроизводительных цифровых ЭВМ факультета электротехники Флоридского университета (США) разработали размещаемое на четырех платах устройство, предназначенное для подключения к компьютерам, совместимым с IBM PC AT. Оно придает этим ПЭВМ архитектуру супер-ЭВМ и повышает на некоторых задачах скорость счета в 2000 раз. Устройство включает в

себя матричный процессор с плавающей точкой, состоящий из 9 микропроцессоров, и контроллер связи с ПЭВМ. Программы для него могут создаваться как человеком (с помощью специальных языков), так и автоматическим преобразователем последовательных алгоритмов с вложенными циклами в параллельные алгоритмы.



## Так ли плох кассетный магнитофон?

Фирма Wangtek (США) объявила о завершении разработки накопителя информации для ЭВМ, использующего магнитофонные кассеты. Емкость одной кассеты — 1,3Г байта, время поиска файла — не более 40 с. Чересчур хорошие параметры? Не удивляйтесь. В накопителе используются кассеты и формат записи данных, разработанные для цифровых кассетных магнитофонов (стандарт DAT).



## Как составить алгоритм

Построение алгоритмов — сложная проблема, требующая от человека высокого уровня логического мышления, творческой деятельности. В нынешнем виде школьный курс информатики больше ориентирован, к сожалению, на заучивание приводимых в учебнике алгоритмов, чем на формирование умения создавать новые. Даже для значительной части учителей, ведущих этот курс, составление алгоритма решения нетипичной задачи — пока еще довольно сложная проблема.

В предлагаемой статье на конкретных примерах рассматриваются два основных приема решения задачи (построения алгоритма): метод моделирования и метод сведения сложной задачи к одной или нескольким простым.

Метод моделирования — это способ переформулировать задачу, заменить ее равносильной, однако более наглядной, что позволяет быстрее найти путь к построению алгоритма. Если «вручную» решить такую задачу несложно, то следует тщательно проанализировать все выполняемые человеком действия, раскладывая их до элементарных. Для того чтобы легче было осуществлять различные способы поиска алгоритма, полезно использовать рисунки, схемы, таблицы и другие наглядные средства.

**Пример 1.** Составить алгоритм, в результате выполнения которого величина  $A$  приняла бы значение величины  $B$ , а  $B$  приняла бы значение величины  $A$ .

*Решение 1.* 1.  $A := B$   
2.  $B := A$

Это решение неверное, так как после первой же команды присваивания значение величины  $A$  будет уничтожено. Вторая команда присваивает величине  $B$  новое текущее значение величины  $A$  (уже равное значению величины  $B$ ).

*Решение 2.* Воспользуемся вспомогательной моделью данной задачи. Пусть  $A$  — это стакан с чаем («стакан» — имя величины, «чай» — ее значение), а  $B$  — чашка кофе («чашка» — имя другой величины, «кофе» — ее значение). Необходимо перелить кофе в стакан, а чай в чашку так, чтобы напитки не смешались.

Теперь становится ясно, что без третьей, вспомогательной емкости (например, кружки) модельную задачу не решить. Имея кружку, последовательность необходимых действий (алгоритм) можно описать так:

1. Содержимое стакана перелить в кружку (кружка := стакан).

2. Содержимое чашки перелить в стакан (стакан := чашка).

3. Содержимое кружки перелить в чашку (чашка := кружка).

Рассмотренная модель — не единственно возможная. Например, можно представить, что  $A$  и  $B$  — это кастрюли, стоящие соответственно на первой и второй конфорках плиты. Вопрос состоит в том, как их переставить. Ясно, что любой человек сразу же предложит отставить в сторону одну из кастрюль (для этого требуется промежуточное место), а вся процедура будет состоять из трех шагов.

Возвращаясь к исходной задаче и вводя вспомогательную (промежуточную) величину  $P$  (такого же типа, как  $A$  и  $B$ ), получим правильное решение:

1.  $P := A$       или      1.  $P := B$   
2.  $A := B$       2.  $B := A$   
3.  $B := P$       3.  $A := P$

*Решение 3.* Если  $A$  и  $B$  — числовые величины, то можно было бы обойтись и без вспомогательной величины  $P$ :

1.  $A := P + B$   
2.  $B := A - B$   
3.  $A := A - B$  (докажите правильность).

Но для литерных  $A$  и  $B$  или настоящих чая и кофе такой фокус не удался бы, т. е. этот вариант решения не универсален. А главное — не понятен. Следует помнить, что ясность алгоритма — одно из наиболее важных его качеств.

**Пример 2.** Вычислить сумму элементов линейной таблицы  $A$  [1:M].

*Решение.* Если бы суммировалось, например, пять чисел, то задачу можно было бы решить одной командой присваивания:

$SUM := A[1] + A[2] + A[3] + A[4] + A[5]$ .

В принципе, так можно было бы поступить и при 500 числах (человек тоже иногда складывает «в столбик» сразу несколько чисел). Но если их количество заранее неизвестно, то одной командой присваивания задачу не решить (ведь в алгоритмах нельзя писать многоточие).

Подберем вспомогательную модель: предположим, что мы суммируем эти числа на микрокалькуляторе (или на счетах). Будем отмечать все действия: включаем МК (если он был вклю-

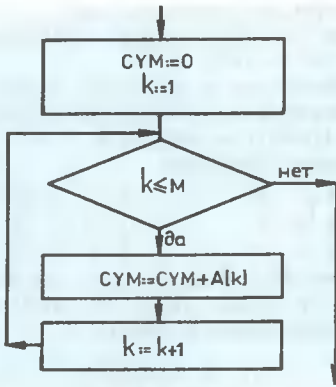
чен, то стираем содержимое индикатора), набираем первое число (можно считать, что мы его прибавили к предыдущему нулю); набираем второе число и добавляем его к содержимому индикатора и т. д. Если через СУМ обозначить содержимое индикатора, то выполняемые действия можно записать следующим образом:

СУМ:= 0  
 СУМ:=A [1] или СУМ:=СУМ+A [1]  
 СУМ:=СУМ+A [2]  
 СУМ:=СУМ+A [3] и т. д.

Теперь ясно, что выполняемые действия состоят в добавлении очередного элемента таблицы А к накапливающейся сумме:

СУМ :=СУМ+А [k], k=1, 2, ..., М.

Блок-схема основной части этого алгоритма выглядит следующим образом:



Точно такие же действия (постепенное накопление результата) будут выполняться и при вычислении произведения  $P=A [1] \cdot A [2] \dots \cdot A [M]$ , степени  $V=x^m$ , факториала  $F!=1 \cdot 2 \cdot \dots \cdot M$  с той лишь разницей, что перед входом в цикл величинам P, V, F нужно присвоить значение 1, а соответствующие команды следует записать в виде:

$P := P \cdot A (k)$   
 $V := V \cdot x,$   
 $F := F \cdot k$

**Пример 3.** Составить алгоритм нахождения значения максимального элемента таблицы X [1:50].

**Решение.** Чтобы облегчить процесс поиска подходящего алгоритма, начнем с рассмотрения частного случая. Пусть  $X=(27, 31, 19, 44, 47, 11, 30)$ . Просмотрев эту таблицу, сразу находим число 47. Следовательно, каким-то алгоритмом поиска максимального элемента мы владеем. Нужно выявить его и записать на алгоритмическом языке.

Анализируем: мы просмотрели все элементы таблицы (значит, алгоритм будет циклическим) и как-то их сравнивали. Что с чем

сравнивали? Может быть, каждый элемент с каждым (тогда будет два вложенных цикла), может быть, два соседних элемента (цикл будет один), но откуда «появляется» максимальный элемент, пока не ясно.

Проверим, достаточно ли одного просмотра таблицы X для выявления максимального элемента. Моделируем ситуацию: закроем таблицу листом бумаги с окошком, в которое можно видеть только одно число, и, перемещая лист слева направо, попытаемся решить задачу (еще лучше, если кто-то будет называть элементы таблицы вслух по одному). Скорее всего, вам удастся найти максимальный элемент (если нет — повторите эксперимент).

Следовательно, за один просмотр массива можно найти нужный нам элемент. А что с чем мы сравнивали, если в окошко в листе бумаги виден только один элемент? Анализируем эксперимент: открыли первое число. Что с ним сделали? Запомнили! Открыли второе число. Сравнили с первым? Но первое число уже не видно. Значит, сравниваем второе число с тем, которое мы запомнили. В результате сравнения может выясниться, что либо второе число больше того, которое мы помним, либо не больше. Если больше, то мы должны его запомнить, в противном случае это число запоминать не нужно (но нужно помнить первое). Открываем третье число. Сравниваем с тем, которое мы помним (!) (совершенно не существенно, первый это или второй элемент), и если третий элемент больше, то запоминаем его.

Итак, алгоритм решения ясен. Чтобы записать его, нужно ввести обозначение для запоминаемого элемента. Пусть имя этой величины будет ПАМЯТЬ. Тогда выполняемые нами действия при поиске максимального элемента на алгоритмическом языке описываются следующим образом:

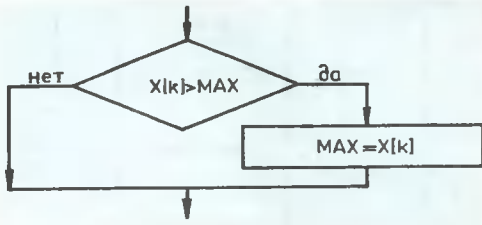
ПАМЯТЬ:=X [1]  
 k:=2  
 пока  $k \leq 50$   
 нц  
   если  $X [k] > \text{ПАМЯТЬ}$   
   то ПАМЯТЬ:=X [k]  
   все  
   k:= [k] + 1

**кц**  
 Конечно, для запоминаемого элемента мы могли бы использовать и другое имя, более подходящее по смыслу например МАКС, но заметим, что имя будет соответствовать смыслу только в правильно составленном алгоритме.

Постепенное приближение к решению задачи нахождения значения максимального элемента таблицы X [1:M] осуществляется сравнением очередного рассматриваемого

элемента  $X[k]$  с максимальным из всех предыдущих:

2



Перед входом в цикл переменной  $MAX$  следует присвоить значение первого элемента, а цикл можно начать с  $k=2$ .

Метод расчленения сложной задачи на более простые, а последних на еще более простые до тех пор, пока не удастся решить все подзадачи, является одним из фундаментальных приемов построения алгоритмов. Этот общий подход получил в программировании название метода последовательного построения (уточнения) или метода проектирования «сверху вниз». Для его реализации на первых этапах построения алгоритма разрешается употреблять команды, которые могут превышать возможности исполнителя, так называемые крупные команды. После последовательной детализации (уточнения) всех крупных команд (возможно снова с помощью разбиения задачи на подзадачи) либо полученные фрагменты алгоритма, состоящие только из команд, входящих в систему команд исполнителя, объединяются в один алгоритм, либо, если задача громоздкая, выделяется основной алгоритм и необходимое количество вспомогательных.

**Пример 4.** Составить алгоритм определения количества различных чисел в таблице цел таб  $A [1:100]$ .

*Решение.* Метод последовательного построения алгоритма позволяет сделать первый шаг в решении задачи (главное на этом шаге — определить аргументы и результаты алгоритма):

алг КОЛИЧ\_РАЗЛ\_ЭЛ\_ТОВ (цел  
таб  $A (1:100)$ , цел  $K$ )  
арг  $A$   
рез  $K$   
нач

Вычисление  $K$  — количества различных элементов в таблице  $A$

кон

Фраза, составляющая этот алгоритм, является крупной командой. Уточнение ее начнем с разбора решения задачи для таблицы с меньшим количеством элементов (используем метод моделирования):

$A (9, 5, 9, 7, 3, 5, 9, 7)$ .

Выяснить, сколько в этой таблице различных элементов, не очень сложно, но нам придется прибегнуть либо к помощи пальцев рук, либо к расстановке пометок. Несомненным является одно: все элементы, таблицы нужно просмотреть хотя бы один раз. Следовательно, в алгоритме будет использоваться команда повторения. Какие действия мы выполняем в цикле? Рассмотрим, например, 3-й элемент таблицы. Поскольку нужно знать количество различных элементов, мы должны выяснить, элемент  $A [3]$  уже был или это новое число и его нужно учесть. Решим «вручную» модельный пример, отмечая символом «+» каждый новый элемент, и символом «-» уже встречавшийся:

$A (9, 5, 9, 7, 3, 5, 9, 7)$   
 + + - + + - - -

Заметим, что первый элемент всегда будет отмечен символом «+» (даже если все элементы таблицы одинаковы). Значит, есть смысл анализировать элементы таблицы, начиная со второго. Итак, крупная команда первоначального алгоритма может быть уточнена следующим образом:

$K := 1$   
 для  $i$  от 2 до 100  
 нц

Если  $A [i]$  — новое число, то  $K := K + 1$

кц

Следующий шаг уточнения алгоритма состоит в разборе новой крупной команды: узнать, встречалось ли значение  $A [i]$  среди предшествующих элементов таблицы. Сформулируем ее в виде новой задачи: литерной величине  $Y$  присвоить значение «новый», если  $A [i]$  не равно ни одному из элементов  $A [1], \dots, A [i-1]$ , и значение «уже был», если равенство выполняется хотя бы один раз. Решение этой задачи опишем в виде вспомогательного алгоритма:

алг ПРОВЕРКА (цел таб  $A [1:i]$ , цел  $i$ , лит  $Y$ )  
арг  $A$ ,  
рез  $Y$   
нач цел  $j$   
 для  $j$  от 1 до  $i-1$   
 нц  
 если  $A [i] = A [j]$   
 то  $Y :=$  «уже был»  
 иначе  $Y :=$  «новый»  
 все  
 кц

кон

Проверим, верно ли работает этот алгоритм, заполняя таблицу исполнения (проверять нужно оба варианта возможного результата).

41

№ шага	i	A		j	v	Проверка условий
		A [1]	A [2]			
1	2	9	5	1	новый	5=9? нет
2						
3						
4						

№ шага	i	A			j	v	Проверка условий
		A [1]	A [2]	A [3]			
1	3	9	5	9	1	уже был	9=9? да
2							
3					2	новый	9=59? нет
4							
5							
6					3	новый	
7							

42

Во втором случае получен неверный результат, причем, после того как было выяснено, что число 9 уже встречалось в таблице. Допущена одна из распространенных ошибок в такого рода задачах: значение «уже был» присваивается Y, если встретится хотя бы один, равный A [i], элемент, но о том, что A [i] — новый элемент, можно говорить лишь после окончания цикла. Типичный пример решения этой задачи:

```

Y:=«новый»; j:=1
пока j<=i-1 и Y=«новый»
  нц
    если A [i]=A [j]
      то Y:=«уже был»
    все
      j:=j+1
  кц
  
```

Заметим, что правильный ответ можно получить и с помощью команды повторения «для», но такой вариант решения будет менее удачен, так как требует полного просмотра всех предшествующих элементов, несмотря на то что вопрос может быть решен уже на первом шаге.

Итак, последняя крупная команда, которую мы выделили в отдельную подзадачу, реализуется вспомогательным алгоритмом:

```

алг ПРОВЕРКА_2 (цел таб A (1:i),
цел i, лит Y)
  арг A, i
  рез Y
нач цел j
  Y:=«новый»; j:=1
  
```

```

пока j<i и Y=«новый»
  нц
    если A [i]=A [j]
      то Y:=«уже был»
    все
      j:=j+1
  кц
кон
  
```

кон

Основной алгоритм теперь будет выглядеть следующим образом:

```

алг КОЛИЧ_РАЗЛ_ЭЛ_ТОВ (цел таб
A [1:100], цел K)
  арг A
  рез K
нач цел i, лит Y
  K:=1
  для i от 2 до 100
    нц
      ПРОВЕРКА_2 (A, i, Y)
      если Y=«новый»
        то K:=K+1
      все
    кц
кон
  
```

кон

Решение поставленной задачи окончилось составлением алгоритма, состоящего из двух частей. Можно, конечно, вместо команды вызова вспомогательного алгоритма вставить в основной алгоритм весь фрагмент вычисления значения величины Y и закончить решение задачи составлением одного большого алгоритма, но такой алгоритм потеряет имеющуюся сейчас ясность.

Подведем некоторые итоги. Прежде чем

составлять алгоритм решения новой задачи, следует попытаться решить эту задачу (или ее упрощенный аналог) вручную, может, быть даже несколько раз, чтобы заметить все действия, выполняемые человеком (как письменно, так и устно — например, проверку условий).

Если задача достаточно сложная, то решение ее будет осуществляться постепенно, с использованием крупных команд. Каждый шаг уточнения алгоритма, как правило, состоит из трех этапов: анализ ситуации (возможно с использованием метода моделирования), построение (синтез)

более точного фрагмента алгоритма, контроль правильности этого фрагмента и его связи с предшествующим.

Завершающим шагом построения алгоритма является, естественно, его проверка в целом. Наиболее простой способ проверки — исполнение алгоритма при специально подобранных вариантах исходных данных (тестовые примеры) с легко определяемым (или заранее известным) ответом. Тестовые примеры должны быть достаточно разнообразны, чтобы можно было проверить все ветви алгоритма.

## Г. СКОБЕЛЕВ

# ПМК на уроках математики

43

Имея ПМК, в школе можно преподавать ОИВТ на технической базе — калькуляторах (машинный вариант), обратив особое внимание на знакомство с алгоритмами решения математических задач и их программирование на ПМК. Это позволит в процессе изучения ОИВТ создать своеобразную библиотеку математических программ (БМП) с последующим ее использованием на уроках математики.

Цели составления и использования библиотеки математических программ:

знакомство с программированием в курсе ОИВТ на базе математического материала; использование ПМК для анализа и лучшего усвоения алгоритмов решения базовых математических задач;

использование ПМК на уроках математики для экономии времени при громоздких вычислениях;

выработка умений и навыков в использовании ПМК.

По курсу алгебры и началам анализа для ПМК «Электроника МК-61» нами разработано 22 программы, реализующие алгоритмы решения основных задач.

1. Решение уравнений:  $ax=b$ ;  $ax^2+bx+c=0$ ;  $ax^3+bx^2+c=0$ ;  $\cos x=a$ ;  $\sin x=a$ ;  $a^x=b$ .

2. Решение неравенств:  $ax>b$ ;  $ax<b$ ;  $ax^2+bx+c>0$ ;  $ax^2+bx+c<0$ ;  $\sin x>a$ ;  $\sin x<a$ ;  $\cos x>a$ ;  $\cos x<a$ ;  $\lg x>a$ ;  $\lg x<a$ ;  $a^x>b$ ;  $a^x<b$ .

3. Вычисление значений функции  $y=f(x)$  на промежутке  $[a, b]$  с шагом  $h$  изменения аргумента.

4. Вычисление площади криволинейной трапеции.

5. Вычисление значений многочлена.

6. Решение системы двух уравнений.

В качестве дополнительных алгоритмов можно рассмотреть: нахождение максимума (минимума) функции; решение уравнений методом деления отрезка пополам и т. д.

Тесные межпредметные связи между ОИВТ и математикой позволяют провести работу над составлением БМП в следующей последовательности.

1. На уроках математики учитель добивается ясного понимания учащимися факта решения любой задачи данного класса с помощью одного и того же алгоритма.

2. Под руководством учителя школьники анализируют структуру алгоритма и составляют его блок-схему. Процесс исполнения алгоритма доводится до автоматизма. Так, например, для решения уравнения  $ax=b$  ученик должен ясно представлять, что, прежде всего, необходимо сравнить  $a$  и  $0$ . Если  $a \neq 0$ , то  $x=b:a$ . Если же  $a=0$ , то сравниваются  $b$  и  $0$ . В случае  $b \neq 0$  решений нет, а при  $b=0$  решением является любое действительное число.

3. На уроках ОИВТ ставится известная из курса математики задача, рассматривается и исследуется алгоритм ее решения (учащимися под руководством учителя), составляется блок-схема алгоритма и анализируется исполнение последнего.

4. Алгоритм реализуется на ПМК, и составленная программа включается в БМП.

5. Составленная программа используется на уроках математики при изучении или повторении соответствующей темы.

Рассмотрим, например, задачу: «Вычислить значения некоторой функции  $y=f(x)$  на про-

межутке  $[a, b]$  с шагом  $h$  изменения аргумента». Эта задача — стандартная, и существует алгоритм, позволяющий решить ее для любой из рассматриваемых в школе функций.

На уроке математики устанавливается, что решение поставленной задачи приводит к составлению таблицы значений данной функции для аргументов, начинающихся с  $a$ , отличающихся друг от друга на  $h$  и расположенных на промежутке  $[a, b]$ . Под руководством учителя учащиеся выделяют шаги алгоритма и их последовательность.

1. Вычислить  $f(x)$  (при  $x=a$ ).

2. Найти  $x:=x+h$ .

3. Проверить условие  $b \geq x$ . Если оно выполняется, то перейти к пункту 1, а иначе — к пункту 4.

4. Окончить вычисления.

Подчеркивается, что сигналом к окончанию работы является невыполнение условия  $b \geq x$ .

44 Затем составляется блок-схема алгоритма и начинается работа по проверке исполнения алгоритма с использованием блок-схемы.

На уроках ОИВТ учитель возвращается к разобранной задаче, анализирует алгоритм ее решения, составляет с помощью школьников блок-схему, проверяет исполнение алгоритма. Ставится вопрос: «Что вызывает наибольшие трудности при составлении таблицы значений функции?» Учащиеся находят очевидный ответ: «Вычисления, требующие большой затраты времени». Поэтому возникает естественный выход из положения — составить программу для ПК и использовать ее для конкретных функций.

Перед составлением программы нужно еще раз вернуться к блок-схеме, выяснить, какой основной структурой реализуется алгоритм (в данном случае циклом). Затем составляется программа для ПК.

Такие программы применяются при исследовании различных функций, при уточнении графиков, при графическом решении уравнений и т. д.

БМП позволяет разнообразить работу над алгоритмами на уроках математики. Например, если в программную память МК-61 введена программа решения уравне-

ния  $\cos x = a$ , то можно предложить такие задания.

1. Решить уравнение  $\cos x = 1,2$ . Почему на индикаторе высвечивается ERROR?

2. Решить уравнение  $\cos x = 0,5$ . Что означает появившееся на индикаторе число 1,0471975? Как записать общее решение данного уравнения?

3. Решить уравнение  $\cos x = 0$ . Что означает показание индикатора 1,5707963? Решением уравнения  $\cos x = 0$  является  $-0,5\pi \approx -1,5707963$ . Нет ли противоречия с показанием индикатора? Как записать общее решение уравнения  $\cos x = 0$ ?

4. Решить уравнение  $\cos x = -0,5$ . Почему на индикаторе высвечивается 2,094395, а не  $-\frac{1}{3}\pi \approx -1,0471975$ ?

БМП позволяет расширить диапазон задач с практическим содержанием. Например, при изучении показательных уравнений можно предложить задачу:

«Количество  $P$  дрожжей, получаемое через  $t$  часов после начала брожения, выражается формулой  $P = P_0 \cdot e^{mt}$ , где  $m$  — некоторый коэффициент. Найти  $m$ , если известно, что через 2 часа после начала брожения прирост дрожжей на 19,1 кг больше прироста дрожжей после 1 часа брожения, а количество дрожжей в начальный момент равно  $P_0 = 90,9$  кг».

Решение задачи сводится к составлению уравнения:

$$90,9e^{2m} - 90,9e^m - 19,1 = 0,$$

которое приводится к виду  $y^2 - y - 0,210 = 0$ . Это уравнение легко решается с помощью программы для решения квадратных уравнений. Найдя  $y \approx 1,1782329$ , из соотношения  $e^m \approx 1,1782329$  находим  $m \approx 0,16401577$ .

Школьный курс математики содержит обширный материал, дающий учащимся возможность формировать, изучать и применять алгоритмы. Тесные межпредметные связи между математикой и ОИВТ и систематическое применение ПК в процессе преподавания этих предметов позволяют сделать обучение школьников алгоритмическому мышлению более продуктивным, а процесс обучения математике — более эффективным.

**А. ГРИЦЕНКО**

зав. кабинетом ОИВТ Алтайского КИУУ

## Информационно-поисковая система для КУВТ-86

Нарастающий в последнее время поток информации привел к тому, что возникла необходимость для обработки больших информационных массивов широко использовать вычислительную технику. Именно поэтому в программное обеспечение любой современной ЭВМ непременно входит информационно-поисковая система (ИПС). Это программа, предназначенная для создания и хранения на магнитных носителях больших массивов справочной информации и быстрого поиска в них требуемых сведений.

Информационный массив, с которым работает ИПС, принято называть базой данных (БД). Базы данных обычно хранят на внешних запоминающих устройствах и по мере надобности работают с ними с помощью специальной программы, которую принято называть системой управления базой данных (СУБД). СУБД позволяет создавать множество БД, дополнять их новыми элементами, изменять информацию, хранящуюся в БД, осуществлять поиск необходимых сведений и т. д.

В ближайшем будущем ИПС будут применяться значительно шире и чаще, чем сегодня, поэтому необходимо как можно более широкое распространение навыков работы с ИПС. Добиться этого — прямая задача курса ОИВТ в средней школе.

Возможные применения ИПС в практике работы средних учебных заведений обстоятельно рассмотрены в [1].

К сожалению, ИПС в школах практически не изучаются даже в тех учебных заведениях, где имеются ЭВМ. Причина тому — отсутствие соответствующего программного обеспечения, недостаточный уровень подготовки преподавательских кадров и отсутствие методической литературы по данному вопросу.

Эти обстоятельства заставили нас приложить значительные усилия для того, чтобы создать ИПС, пригодную для эксплуатации в учебном процессе на КУВТ-86.

Работа была завершена в марте 1988 г. Данная статья посвящается описанию этой системы и рекомендациям по ее использованию в учебном процессе и является фактически продолжением статьи [2].

### Состав ИПС

Предлагаемая система состоит из следующих частей:

BASE — программа СУБД;

BDCON — учебная база данных «Страны мира», содержит сведения о всех существующих в мире государствах и зависимых территориях;

EDASP — экранный редактор текстов для БК-0010, используется для создания собственных БД;

BASE.TXT — текстовый файл с описанием СУБД BASE;

EDASP.TXT — текстовый файл с описанием экранного редактора EDASP;

NETF.SAV — программа связи для классов с языком FOCAL.

Программы BASE и EDASP были созданы в 1987 г. А. В. Гречишкиным, С. А. Фоменко и П. В. Успенским. Эти программы имели следующие существенные для КУВТ-86 недостатки:

были рассчитаны на работу только с одноканальным БК;

допускали ввод-вывод только на магнитофон.

Поэтому пришлось серьезно их переработать, чтобы они могли:

пересылаться по сети КУВТ-86 (как обычные кодовые программы);

осуществлять прямое обращение БК-0010 к дисководу ДВК-2 для ввода-вывода данных.

Эта работа была проделана О. О. Войцеховичем (г. Барнаул) при участии автора статьи. Им же создана программа NETF.SAV. База данных BDCOW создана автором статьи по материалам справочника [3].

Информацию о данной ИПС и об условиях ее приобретения можно получить по адресу: 656099, г. Барнаул, пр. Социалистический, 60, каб. № 15.

### Начало работы с СУБД BASE

СУБД BASE пересылается по сети КУВТ-86 как обычная кодовая программа. После пересылки происходит автозапуск программы и на экране появляется список функций (меню), которые может выполнять СУБД BASE:

- Считывание файла
- Поиск
- Редактирование (1)
- Дополнение
- Запись файла
- Выход

Одна из этих строчек будет подчеркнута прямой линией. Так выделяется функция, которую будет выполнять СУБД после нажатия клавиши ВВОД. Местоположение этой черты-указателя можно изменять с помощью клавиш «стрелка вверх» и «стрелка вниз», выбирая таким образом ту функцию СУБД, которую ей следует исполнить.

После загрузки СУБД необходимо считать в нее находящийся на диске информационный файл, содержащий базу данных (БД). Для этого нужно установить указатель на строку, где указана функция «Чтение файла», и нажать клавишу ВВОД. После этого СУБД очистит экран и выдаст запрос:

Считать файл? (2)

На это нужно ответить нажатием клавиши Д (да), и на экране появится следующий запрос:

Введите имя файла: (3)

В ответ нужно ввести имя файла, содержащего БД, с которой предполагается работать. Таким именем может быть, например, BDCON — имя БД «Страны мира». Набрав имя файла, следует нажать клавишу ВВОД.

СУБД осуществляет обращение по сети к дисководу КУВТ-86. При этом происходит поиск на диске файла с указанным именем. Если такого файла на диске не оказалось, что бывает, например, если имя файла набрано с ошибкой, то на экране появится сообщение

Ошибка чтения (4)

а затем

Считать следующий файл? (5)

Если в ответ нажать клавишу Д, то опять появится запрос (3) и т. д.

Если же файл с указанным именем найден на диске, происходит его чтение в память БК. Иногда при чтении файла с дисковода возникает ошибка. В этом случае опять появляются запросы (4) и (5) и т. д. Если это произошло, нужно повторить чтение. Если же и это не помогает, то файл, содержащий БД, испорчен и требуется его перезапись. Но это (при исправном дисковом) случается крайне редко.

Если файл считан без ошибок, на экране появится запрос (5). На него следует ответить нажатием клавиши Д в том случае, если БД, с которой предполагается работать, состоит не из одного, а из нескольких файлов. При этом вновь появится запрос (3) и весь процесс чтения начнется снова. Все файлы, которые последовательно считываются, будут состыкованы в памяти БК в одну БД.

Если же читать следующий файл не требуется, то на запрос (5) нужно ответить нажатием клавиши Н (нет) и СУБД выйдет в свое основное меню (1).

### Режим «Поиск»

Чтобы перейти в основной режим работы СУБД «Поиск», необходимо в меню (1) установить указатель и нажать ВВОД.

Если до перехода в этот режим не происходило чтение информационного файла, на экране появится сообщение

Нет данных

и по нажатию любой клавиши произойдет возврат в основное меню (1).

Если в память БК был считан файл, который не является специально подготовленной для данной СУБД базой данных, то при попытке перейти в режим поиска на экране появится сообщение

Данные не в формате BASE

и по нажатию любой клавиши произойдет возврат в основное меню (1).

Если же считанный в СУБД файл удовлетворяет ее требованиям, т. е. является специально для нее подготовленной базой данных, то после входа в режим «Поиск» на экране появится список имен ключевых признаков, по которым в данной БД можно производить поиск информации.

Например, для БД BDCON этот список будет иметь следующий вид:



Страна :  
 Часть света :  
 Столица :  
 Население :  
 Общ. строй :  
 Гос. язык :

Курсор будет стоять в первой из этих строк после двоеточия.

После появления на экране этого списка следует задать в каждой из его строк значение ключевого признака и нажать ВВОД. Программа СУБД осуществит поиск в памяти БК информации, обладающей всеми указанными признаками, и выведет ее на экран в виде таблицы.

Ключевой признак — это последовательность из нескольких символов. Набрав его в строке, следует клавишей «стрелка вниз» переместить курсор на следующую строку, и можно набирать следующий ключевой признак. Задаваемые пользователем ключевые признаки могут быть и пустыми, т. е. вообще не содержать символов. Чтобы задать такой ключевой признак, необходимо поставив курсор в соответствующую строку, нажать, ничего в ней не набирая, на клавишу «стрелка вниз» или ВВОД. В последнем случае начнется процесс поиска, а все признаки, задаваемые в строках, лежащих ниже курсора, будут считаться пустыми.

Алгоритм, с помощью которого СУБД осуществляет поиск информации, прост.

1. Перебираются все находящиеся в памяти БК информационные строки и выбираются те из них, в которых в соответствующих позициях имеются указанные пользователем сочетания символов, т. е. непустые ключевые признаки.

2. Найденные таким образом строки из БД высвечиваются на экране дисплея в виде таблицы.

3. Если поиск закончен и для найденных строк хватило места на экране, то под таблицей появляется сообщение

Поиск закончен (7)

и по нажатию любой клавиши происходит возврат в основное меню (1).

4. Если просмотр всей БД не закончен, а на экране уже нет свободных строк, то поиск приостанавливается и под таблицей появляется сообщение

Продолжать поиск? (8)

Если в качестве ответа нажать клавишу N, то происходит возврат в основное меню (1). Если в качестве ответа нажать любую другую клавишу, то поиск продолжается и таблица на экране заполняется новыми найденными строками.

Если СУБД не смогла найти во введенной в память БД ни одной строки, в которой бы содержались все указанные пользователем ключевые признаки, то на экране появится таблица, содержащая лишь пустую строку и сообщение (7).

При поиске СУБД производит проверку имеющихся в памяти информационных строк только на наличие в них непустых ключевых признаков. Поэтому если какие-то из ключевых признаков заданы пустыми, то они при поиске не играют никакой роли.

Это свойство можно использовать для организации поиска по любой группе признаков. Например, при работе с БД BDCON поиск с набором ключевых признаков вида

Страна : Австрия  
 Часть света :  
 Столица :  
 Население :  
 Общ. строй :  
 Гос. язык :

47

приведет к выдаче на экран всех указанных сведений об Австрии, оформленных в виде таблицы. Задав пустыми все ключи, можно просмотреть БД целиком. Это полезно сделать при знакомстве с новой БД, чтобы ознакомиться с ее составом и принятыми сокращениями. Так, например, в БД BDCON названия частей света хранятся в сокращенном до двух букв (прописных) виде, т. е. EB, AZ, AF, SA, YA, AB означают соответственно Европа, Азия, Африка, Северная и Южная Америка, Австралия. Если не знать этих сокращений, то осуществить поиск по данному признаку не удастся.

При задании ключевых признаков имеют-ся дополнительные возможности:

символ «а» заменяет любой символ;

символ «○» заменяет любую последовательность символов;

символ «—» означает, что поиск по данному признаку происходит не по совпадению его с соответствующей позицией информационной строки, а наоборот, по отсутствию такого совпадения.

Например, выполнение поиска в БД BDCON с таким набором ключевых признаков:

Страна : A○  
 Часть света : —AF  
 Столица : B а а а  
 Население :  
 Общ. строй :  
 Гос. язык :

заставит программу СУБД выдать на экран

сведения о всех странах мира, названия которых начинаются на букву «А», находящихся не в Африке, названия столиц которых состоят из четырех букв и начинаются с буквы «В».

Символ «␣» удобен для экономии времени и усилий при вводе ключевых признаков. Так, вместо слова «немецкий» можно задать «нем␣», так как название только одного государственного языка начинается с букв «нем».

### Режим «Дополнение»

СУБД BASE позволяет заносить в БД новые информационные строки. Для этого предназначен режим «Дополнение». После входа в него (из меню (1)) на экране дисплея появится список ключевых признаков, по которым в данной БД можно проводить поиск информации (как в режиме «Поиск»). Затем нужно набрать в каждой из строк этого списка необходимую информацию, которая и образует добавляемую в БД новую информационную строку. При этом можно использовать клавиши управления курсором «стрелка влево», «стрелка вправо», «стрелка вниз», «возврат курсора со стиранием символа». Возврат курсора вверх на набранную ранее строку не допускается. Попытка сделать это приводит к выходу в меню (1) без всяких изменений в БД. После завершения ввода данных следует нажать клавишу ВВОД. В БД будет занесена новая информационная строка, а СУБД выйдет в меню (1).

Дополненную БД целесообразно сразу же записать на диск, используя режим «Запись файла». В противном случае дополнения будут утеряны после выключения данного БК.

### Режим «Редактирование»

Этот режим предназначен для изменения или удаления информационных строк. После входа в него (из меню (1)) на экране появляется список ключевых признаков.

Прежде чем отредактировать информационную строку, ее нужно найти. Для этого нужно ввести ключевые признаки, обеспечивающие однозначное нахождение именно этой строки режимом «Поиск» (иногда бывает полезно предварительно «отладить» этот набор признаков в режиме «Поиск») и нажать клавишу ВВОД. Найденная информационная строка появится на экране, после чего можно вносить в нее изменения, пользуясь клавишами управления курсором, как в режиме «Дополнение». Окончив редактирование, следует нажать клавишу ВВОД. Информационная строка запишется в память в отредактированном виде.

Если вместо редактирования нажать клавишу СТОП, то указанная информационная строка будет удалена из базы данных.

После редактирования, как и после дополнения, рекомендуется записать новый вариант БД на диск.

### Режим «Запись файла»

Этот режим применяется для записи на диск дополненной или отредактированной базы данных. После входа в него на экране появляется запрос

#### Введите имя файла:

В ответ нужно набрать имя (любую последовательность из не более чем шести заглавных латинских букв и цифр), под которым новую базу данных следует записать на диск. В данной системе сложилась традиция: две первые буквы имени любой базы данных — ВД. Благодаря этому в каталоге сразу видно, какие файлы являются базами данных.

После набора имени следует нажать на клавишу ВВОД, и вся имеющаяся в памяти БК БД будет записана на диск. При этом на ДВК должна работать программа, обрабатывающая запросы на обращение к диску с рабочих мест учащихся (это NET.SAV для КУВТ-86 с Бейсиком и NETF.SAV для КУВТ-86 с Фокалом).

От записи можно отказаться, нажав клавишу СТОП.

После завершения записи на диск или отказа от нее программа СУБД вернется в меню (1).

### Режим «Выход»

Это — режим прекращения работы с СУБД BASE. Вход в него приводит к появлению запроса

Выход, Вы уверены?

Если в ответ нажать клавишу Д, то программа прекратит свою работу. Если нажать любую другую клавишу, то произойдет выход в меню (1).

### Создание новой БД

Прежде чем создавать новую БД, следует решить:

какую информацию будут содержать информационные строки;

каков будет список ключевых признаков;

каковы будут заголовок и формат таблицы, выводимой на экран дисплея;

какие будут использованы обозначения и сокращения;

какое имя будет у новой БД.

Существенным недостатком рассматриваемой ИПС является требование, чтобы каждая информационная строка умещалась в строке экрана дисплея БК, т. е. имела длину не более 64 символов. Это и заставляет применять условные обозначения и сокращения.

Режим «Дополнение» позволяет лишь расширять уже существующие БД. Для создания новой БД необходимо сначала сделать ее заготовку с помощью экранного редактора текстов EDASP, описанного в [2]. Именно для этого EDASP и включен в данную ИПС.

Лучший способ узнать, как делается заготовка БД,— считать с помощью EDASP какую-либо ранее созданную БД как обычный текстовый файл. На экране дисплея БК станет видна внутренняя структура файла БД. Сразу станет видно, что этот файл состоит из:

списка ключевых признаков, который выводится на экран при использовании режима «Поиск»;

заголовка и верхней части таблицы, которая выводится на экран при выполнении поиска;

черты, ограничивающей эту таблицу снизу.

Эти компоненты разделяются строками, содержащими единственный символ «перевод формата» в первой позиции. Этот символ выглядит на экране как прямоугольник и набирается нажатием клавиш СБР или СУ/Л. Далее следуют сами информационные строки, составляющие БД.

Чтобы создать заготовку БД, следует с помощью EDASP набрать соответствующие компоненты будущей БД и записать полученный текстовый файл на диск с помощью соответствующей команды EDASP.

Например, для создания БД BDCON пришлось сначала создать следующий текстовый файл.

Страна :  
 Часть света :  
 Столица :  
 Население :  
 Общ. строй :  
 Гос. язык :

#### База данных «Страны мира»

Используемые обозначения:

ЕВ — Европа

АЗ — Азия

АФ — Африка

СА — Северная Америка

ЮА — Южная Америка

АВ — Австралия

Страна	Часть света	Столица	Население	Общ. строй	Гос. язык
--------	-------------	---------	-----------	------------	-----------

Символы, составляющие верхнюю и нижнюю черту таблицы и разделители ее колонок, набираются как псевдографические символы с клавиатуры БК.

Записанную на диск заготовку БД следует считать в память БК, используя режим «Считывание файла» СУБД BASE. После этого можно приступить к наполнению новой БД, используя для этого режим «Дополнение».

#### Рекомендации по методике

Изучение ИПС должно стать неотъемлемой частью курса ОИВТ. В частности, если используется КУВТ-86, то рекомендуем применять вышеописанную ИПС и спланировать занятия по ее изучению следующим образом.

Занятие 1. Лекция. Понятие о ИПС, СУБД и БД. Состав и работа ИПС для КУВТ-86. Режимы работы СУБД BASE. Примеры задач на поиск в какой-либо реальной БД (например, BDCON).

Занятие 2. Чтение с диска заранее подготовленной БД (например, BDCON). Решение содержательных задач на поиск информации в БД.

Занятие 3. Решение содержательных задач на редактирование и дополнение заранее подготовленной БД.

Занятие 4. Создание простейшей новой БД.

Занятие 5, зачетное. Решение задач, включающих чтение БД с диска, поиск информации в ней, редактирование и дополнение ее.

На всех занятиях рекомендуется ставить и решать задачи содержательного характера, например: «Найти все англоязычные страны Африки» или «В Парагвае недавно свергнута военная диктатура; проверьте, отражено ли это в нашей БД, и если нет, то внесите в нее необходимые изменения».

В качестве задач для внеклассной работы можно рекомендовать создание больших БД по разным учебным предметам (химии, физике, биологии и т. д.).

Желаем успеха всем пользователям системы!

М. АЛЕКСЕЕВ, Т. АЛЕКСЕЕВА  
г. Миасс

## Твои первые программы

— Ты не сумеешь, — сказал Торопыжка. — Это ведь машина. Тут понимать надо.

— Чего тут еще понимать! — ответил Незнайка. — Я видел, как ты управляешь. Дергай за ручки да верти руль. Все просто.

Н. Носов.

«Приключения Незнайки и его друзей».

### 0. Введение

Эта статья написана по материалам двух циклов занятий кружка юных программистов на персональных ЭВМ «Агат» в ЦШ № 18 г. Миасса. На занятии обычно разбираются новые понятия в приложении к простому сценарию работы с машиной, у доски составляется программа, а затем ребята вводят ее в свои компьютеры, отлаживая, изменяя и усложняя. Набравшись опыта, учащиеся приступают к составлению сложных игровых и обучающих программ.

В 1986/87 учебном году изучался Бейсик, а в 1987/88 — на тех же сценариях — Рапира. В результате симпатии авторов полностью перешли к Рапире, но сильные и слабые стороны есть, конечно, у обоих подходов. Бейсик очень прост и легок для начинающих, но, к сожалению, этот неструктурный и непроцедурный язык идейно ограничен и не пригоден для серьезной работы. По Бейсику очень много книг и статей в популярных журналах, он есть на многих ЭВМ, но система программирования Бейсик-Агат все же менее комфортабельна, чем система «Школьница» с Рапирой. В отличие от английских ключевых слов Бейсика русские команды «Школьницы» доступны учащимся любого возраста. Классические структуры органично пронизывают любую программу на Рапире в лучших традициях структурного программирования, богатые возможности предоставляет такой динамический тип данных, как кортежи. Про-

## Литература

1. Пахомова Н., Шрейбер Б. Т. База данных школы // Информатика и образование. 1988. № 5.
2. Гриценко А. Н. Система подготовки текстов для КУВТ-86 // Информатика и образование. 1989. № 3.
3. Страны мира: Краткий политико-экономический справочник. М.: Политиздат, 1988.

цедурный подход позволяет организовать разделение программистского труда и создавать сложные пакеты программ. Однако память машины небезразмерна, более мощная «Школьница» занимает по сравнению с Бейсиком больше места в памяти и на диске, соответственно меньше места остается под программы и графику, чаще обмены с дисководом — самым слабым местом «Агата».

В этом пособии программы описаны так же, как они изучались на занятиях кружка. Сначала кратко излагается сценарий (С), затем тексты программ на Рапире и Бейсике и комментарии к ключевым местам в них (К) и, наконец, идет развитие темы (Р) — что еще интересного можно сделать на основе этих программ. Правила записи и смысл операторов Рапиры и Бейсика, приемы работы с машиной здесь не приводятся. Эти вопросы следует прояснять по соответствующим руководствам параллельно с разбором программ. Программы как раз и выстроены в некоторой методической последовательности освоения начал программирования в полугривовой постановке. Естественно, что при этом не все вопросы оказались одинаково подробно рассмотрены. Предназначенное в первую очередь для учителя, это пособие может быть использовано при сравнительном изучении Рапиры и Бейсика и, конечно же, для самостоятельной работы новичков, пришедших в кружок в середине учебного года.

Большое участие в постановке и отладке приведенного программного материала приняли А. Н. Рябова и В. Г. Струц. Использованы также материалы, любезно предоставленные Н. А. Юнерман и А. Ю. Кривцовым.

### 1. Привет!

С. Машина задает вопросы, человек отвечает. Полученную в ответах информацию машина использует в новых вопросах.

```

1 ПРОЦ ПРИВЕТ;
2 ВЫВОД: "ЗДРАВСТВУЙ, МЕНЯ ЗОВУТ
          АГАТ !";
3 ВЫВОД: "А ТЕБЯ ?";
4 ВВОД: ИМЯ;
5 ВЫВОД БПС: "СКОЛЬКО ТЕБЕ ЛЕТ, "; ИМЯ;
6 ВВОД ДАННЫХ: ВОЗРАСТ;
7 ВОЗРАСТ-6->КЛАСС;
8 ВЫВОД: "ПОХОЖЕ, ТЫ УЧИШЬСЯ В", КЛАСС,
          "-М КЛАССЕ";
9 КНЦ;

```

```

10 REM ПРИВЕТ
20 PRINT "ЗДРАВСТВУЙ, МЕНЯ ЗОВУТ АГАТ !"
30 PRINT "А ТЕБЯ ?"
40 INPUT NAME$
50 PRINT "СКОЛЬКО ТЕБЕ ЛЕТ, "; NAME$;
60 INPUT AGE
70 CLASS = AGE - 6
80 PRINT "ПОХОЖЕ, ТЫ УЧИШЬСЯ В "; CLASS;
          "-М КЛАССЕ"

```

К. В строке 7 (70) в предположении, что в 0-м классе свойственно учиться шестилеткам, по возрасту вычисляется класс.

Р. Предусмотреть возможность беседы с вундеркиндом-дошкольником, со взрослым, завести разговор о погоде, о здоровье, о вкусах. Построить программу опроса по какому-либо предмету. Запрограммировать какой-нибудь психологический тест.

## 2. Угадай

С. Машина загадывает целое число от 1 до 100 и предлагает угадать его, оценивая попытки сообщениями типа «перелет — недолет — попал».

```

1 ПРОЦ УГАДАЙ;
2 ?"УГАДАЙ ЦЕЛОЕ ЧИСЛО ОТ 1 ДО 100";
3 ЦЕЛЧ(ДСЧ()*100)+1->N; 0->N;
4 ПОКА N/=N ::
5   ВВОД ДАННЫХ: N;
6   ЕСЛИ N>N TO ?"МЕНЬШЕ" ВСЕ;
7   ЕСЛИ N<N TO ?"БОЛЬШЕ" ВСЕ
8 ВСЕ;
9 ?"МОЛОДЕЦ !!!";
10 КНЦ;

```

```

10 REM УГАДАЙ
20 ?"УГАДАЙТЕ ЦЕЛОЕ ЧИСЛО ОТ 1 ДО 100"
30 N% = RND (1) * 100 + 1
40 INPUT N%
50 IF N% > N% THEN ?"МЕНЬШЕ": GOTO 40
60 IF N% < N% THEN ?"БОЛЬШЕ": GOTO 40
70 ?"МОЛОДЕЦ !!!"

```

К. В строке 3 (30) с помощью датчика чисел ДСЧ (RND), вырабатывающего случайное число в интервале (0; 1), вычисляется загадываемое машиной целое число от 1 до 100.

Р. Можно изменять границы загадывания, очень полезно ввести счетчик попыток. Поиграв с программой, можно выработать оптимальную стратегию угадывания и вывести формулу, позволяющую по диапазону, в котором загадано число, рассчитать коли-

чество попыток, гарантирующее угадывание. Тогда программа сможет оценивать умение угадывать. Можно попробовать поиграть не в «больше — меньше», а в «горячо — холодно» или «близко — далеко» («ближе — дальше»), когда машина следит за абсолютной величиной отклонения от цели. «Горячо — холодно» можно отражать разными цветами.

## 3. Загадай

С. Теперь человек загадывает, а машина угадывает, ориентируясь на его подсказки. При этом используется стратегия половинного деления (дихотомия), «изобретенная» в игре УГАДАЙ.

```

1 ПРОЦ ЗАГАДАЙ;
2 ?"ЗАГАДАЙ ЦЕЛОЕ ЧИСЛО ОТ 1 ДО 100";
3 "ОТВЕЧАЙ: Б-БОЛЬШЕ,М-МЕНЬШЕ,Д-ДА"->
  ИНСТРУКЦИЯ;
4 ?ИНСТРУКЦИЯ; 0->ЛГ; 101->ПГ; ""->ОТВЕТ;
5 ПОКА ПГ-ЛГ>1 ::
6   (ЛГ+ПГ)/2->Н;
7   ?"Это", Н; КЛАВ()->ОТВЕТ;
8   ВЫБОР ОТВЕТ ИЗ
9     "Б": Н->ЛГ;
10    "М": Н->ПГ;
11    "Д": ?"УРА !!!"; ВЫХОД
12   ИНАЧЕ ?ИНСТРУКЦИЯ
13   ВСЕ
14 ВСЕ;
15 ?"ТЫ ОШИБСЯ ИЛИ СХИТРИЛ";
16 КНЦ;

```

```

10 REM ЗАГАДАЙ
20 ?"ЗАГАДАЙ ЦЕЛОЕ ЧИСЛО ОТ 1 ДО 100"
30 I$="ОТВЕЧАЙ: Б-БОЛЬШЕ,М-МЕНЬШЕ,Д-ДА"
40 ?I$:L% = 0:R% = 101
50 N% = (L% + R%) / 2
60 ?"ЭТО "I$N%: GET R$
70 IF R$ = "Б" THEN L% = N%: GOTO 110
80 IF R$ = "М" THEN R% = N%: GOTO 110
90 IF R$ = "Д" THEN ?"УРА !!!": END
100 ?I$: GOTO 60
110 IF R% - L% > 1 THEN 50
120 ?"ТЫ ОШИБСЯ ИЛИ СХИТРИЛ"

```

К. В каждой попытке угадываемое число вычисляется как среднее арифметическое левой и правой границ, в случае «перелета» это значение принимает новая правая граница, а при «недолете» — новая левая, и так до попадания. В программе есть элементы того, что называют «защита от дурака» (foolproof). Если при ответе была нажата не та клавиша, в строке 12 (100) на экран выводится ИНСТРУКЦИЯ (I\$) и запрос повторяется. Если левая и правая границы сблизились вплотную, а играющий не признается, то во избежание топтания на месте программа выдает соответствующее сообщение и прекращает работу. Нормальное завершение здесь не в конце программы, в строке 11 (90).

Р. Здесь тоже пригодится счетчик попыток.

Попробуйте делить не попалам, а в отношении  $(\sqrt{5}+1)/2$  (золотое сечение) и сравните результаты.

#### 4. Обложка

С. Напечатать на принтере «титульный лист», который можно наклеить на тетрадь по информатике.

```

1 ПРОЦ ОБЛОЖКА;
2 ?" ВВЕДИТЕ :";
3 ?"ФАМИЛИЮ,ИМЯ В РОДИТЕЛЬНОМ ПАДЕЖЕ";
4 ?" НАПРИМЕР: ИВАНОВОЙ ОЛЬГИ";
5 ВВОД: ИМЯ;
6 ВЫВОД БПС:"КЛАСС, БУКВУ ";
7 ВВОД: КЛАСС;
8 ВЫВОД БПС:"ВЫ УЧЕНИЦА (Д/Н)";
9 КЛАВ()->ОТВЕТ;
10 ЕСЛИ ОТВЕТ="Д" ТО "ЦЫ "->К
11 ИНАЧЕ "КА "->К
12 ВСЕ;
13 ВКЛ ВЫВОД НА БУМАГУ;
14 ?" Т Е Т Р А Д Ь";
15 ?" ПО ИНФОРМАТИКЕ";
16 ?" УЧЕНИ" ,К,КЛАСС," КЛАССА";
17 ?"СРЕДНЕЙ ШКОЛЫ N 18 Г.МИАССА";
18 ?" ",ИМЯ;
19 ВЫКЛ ВЫВОД НА БУМАГУ;
20 КНЦ;

```

```

10 REM ОБЛОЖКА
20 ?" ВВЕДИТЕ : "
30 ?"ФАМИЛИЮ,ИМЯ В РОДИТЕЛЬНОМ ПАДЕЖЕ"
40 ?" НАПРИМЕР: ПЕТРОВА СЕРГЕЯ"
50 INPUT NAME$
60 INPUT "КЛАСС, БУКВУ "; CLASS$
70 ?"ВЫ УЧЕНИЦА (Д/Н)";
80 GET R$
90 K$="КА "
100 IF R$="Д" OR R$="D" THEN K$="ЦЫ "
110 ? : ? CHR$(4);"PR#5"
120 ?" Т Е Т Р А Д Ь"
130 ?" ПО ИНФОРМАТИКЕ"
140 ?" УЧЕНИ";K$; CLASS$;" КЛАССА"
150 ?"СРЕДНЕЙ ШКОЛЫ N 18 Г.МИАССА"
160 ?" "; NAME$
170 ? : ? CHR$(4);"PR#0"

```

К. В строках 8—12 уточняется, кому — ученику или ученице — печатать обложку. Вывод на принтер включается в строке 13 (110) и выключается в строке 19 (170). Перед печатью необходимо убедиться в том, что принтер подключен к машине и к сети, загружен соответствующий драйвер печати, т. е. все сделано в соответствии с инструкцией по работе с принтером.

Р. Предусмотрите возможность печати обложек для разных предметов. Сделайте автоматическое центрирование, т. е. красивое расположение на равных расстояниях от краев строк с названием предмета и именем владельца тетради, имеющих переменную длину. Изготовьте себе визитную карточку.

#### 5. Рамка

С. «Культурные» программы, как правило,

имеют заставки, содержащие название программы, имена авторов, инструкции. Заставку обычно помещают в рамку.

```

1 ПРОЦ РАМКА;
2 РЖМ("ТЮ"); "Ж"->С;
3 ДЛЯ К ОТ 1 ДО 30 ::
4 ПОЗ(1,К); ВЫВОД БПС:С;
5 ПОЗ(К,1); ВЫВОД БПС:С;
6 ПОЗ(30,К); ВЫВОД БПС:С;
7 ПОЗ(К,30); ВЫВОД БПС:С;
8 ВСЕ;
9 ПОЗ(0,0);
10 КНЦ;

10 REM РАМКА
20 HOME : S$="* "
30 FOR K = 1 TO 30
40 HTAB 1: VTAB K: ?S$;
50 HTAB K: VTAB 1: ?S$;
60 HTAB 30: VTAB K: ?S$;
70 HTAB K: VTAB 30: ?S$;
80 NEXT K

```

К. В строке 2 (20) делается очистка экрана, а затем назначается символ для рамки. Затем в цикле 30 раз по 4 указываются горизонтальные и вертикальные координаты позиции, где печатается символ.

Р. В качестве символа рамки хорош универсальный пробел. Можно построить рамку, переливающуюся разными цветами, несколько вложенных рамок.

#### 6. Покупка

С. Смоделируем процесс купли-продажи. Для простоты начнем с одного товара (пирожных) и одного покупателя. В ходе торговли будем подсчитывать остаток денег у покупателя и выручку у продавца, а также отображать купленные и оставшиеся пирожные. Предусмотрим возможности исчерпания товара или денег.

```

1 ПРОЦ ПОКУПКА;
2 22->ЦЕНА; 40->ПИРОЖНЫЕ; 0->ВЫРУЧКА;
3 ОКНО(0,0,31,31); ?;
4 ?"ПИРОЖНОЕ СТОИТ",ЦЕНА," КОПЕЙКИ";
5 ?;ВЫВОД БПС:"СКОЛЬКО У ВАС КОПЕЕК ";
6 ВВОД ДАННЫХ:ДЕНЬГИ;
7 ?;?" П - ПОКУПКА, К - ВЫХОД";
8 ?;?" УПЛАЧЕНО ОСТАЛОСЬ";
9 ДЛЯ С ОТ 21 ДО 7 ШАГ -2 ::
10 ПОЗ(2,С);
11 ПОВТОР 5 :: ВЫВОД БПС:"'В 'А " ВСЕ
12 ВСЕ;
13 0->П; 21->С; ""->КЛ;
14 ПОКА НЕ КЛ="К" ::
15 КЛАВ()->ЖЛ;
16 ЕСЛИ КЛ="П" ТО
17 ВЫБОР ИЗ
18 ДЕНЬГИ<ЦЕНА: ПОЗ(1,2);
19 ?"ДЕНЕГ НЕ ХВАТАЕТ";ВЫХОД :
20 ПИРОЖНЫЕ=0: ПОЗ(1,2);
21 ?"ПИРОЖНЫЕ КОНЧИЛИСЬ";ВЫХОД
22 ИНАЧЕ
23 ПИРОЖНЫЕ-1->ПИРОЖНЫЕ;
24 ДЕНЬГИ-ЦЕНА->ДЕНЬГИ;

```

```

25 ВВРУЧКА+ЦЕНА->ВВРУЧКА;
26 ПОЗ(6,23);?ВВРУЧКА," ",
        ДЕНЬГИ;
27 П+2->П;
28 ЕСЛИ (П>10 ТО 2->П; С-2->С ВСЕ;
29 ПОЗ(П,С);ВВВОД БПС:" ";
30 ПОЗ(П+15,С);ВВВОД БПС:"'В 'А"
31 ВСЕ
32 ВСЕ
33 ВСЕ;
34 КНЦ;

```

```

10 REM ПОКУПКА
20 Р = 22:С = 40:В = 0: ТЕХТ = 2: НОМЕ
30 ?"ПИРОЖНОЕ СТОИТ ";Р;" КОПЕЙКИ"
40 ? : INPUT "СКОЛЬКО У ВАС КОПЕЕК ";М
50 ? : ?"П - ПОКУПКА, К - ВВХОД"
60 ? : ?" УПЛАЧЕНО ОСТАЛОСЬ"
70 FOR S = 11 TO 25 STEP 2: VТAB S:
        НТAB 2
80 FOR L = 1 TO 5
90 INVERSE : ? " ";: NORMAL : ? " ";
100 NEXT L: NEXT S
110 L = 0: S = 11
120 GET КВ
130 IF КВ = "К" THEN END
140 IF NOT (КВ="П" OR КВ="Р") THEN 120
150 VТAB 30: НТAB 1
160 IF М<Р THEN ?"ДЕНЕГ НЕ ХВАТАЕТ":
        END
170 IF С=0 THEN ?"ПИРОЖНЫЕ КОНЧИЛИСЬ":
        END
180 С=С-1: М=М-Р: В=В+Р: VТAB 9: НТAB 6
190 ? " ";В;" ";М;" "; L=L+2
200 IF L > 10 THEN L = 2: S = S + 2
210 НТAB L: VТAB S: ? " ";
220 НТAB L+15: INVERSE : ? " ";: NORMAL
230 GOTO 120

```

К. После ввода начального количества денег программа управляет двумя клавишами П и К. Слева инверсными пробелами показаны продающиеся пирожные, по ходу они стираются и появляются справа — среди купленных. В распечатке программы на Рапире комбинациям соответствуют: 'L — УПР+Л — очистка экрана, 'В — функциональная точка — инверсный режим, 'А — функциональный ноль — нормальный режим. В «Редакторе» эти комбинации вводятся после нажатия клавиш УПР+Ж.

Р. Естественное развитие темы — разноцветный ассортимент товаров, небольшая очередь покупателей, вежливые вопросы продавца, реклама и бухгалтерия на уровне западного супермаркета.

## 7. Мозаика

С. В режиме цветной графики среднего разрешения в бесконечном цикле случайным цветом ставится точка со случайными координатами.

```

1 ПРОЦ МОЗАИКА;
2 РЖМ("СГОП");
3 ПОКА I=1 : :
4 ЦВЕТ(ДСЧ()*7+1);

```

```

5 ТЧК(ДСЧ()*128,ДСЧ()*128);
6 ВСЕ;
7 КНЦ;

```

```

10 REM МОЗАИКА
20 MGR=1
30 COLOR = RND (1) * 7 + 1
40 X = RND (1) * 128
50 Y = RND (1) * 128
60 PLOT X,Y
70 GOTO 30

```

Р. Можно поэкспериментировать со случайными лучами из одной точки, например из центра экрана, или со случайными отрезками по всему экрану. Попробовать строить при этом симметричные картины в двух половинах или четырех четвертях экрана.

## 8. Муары

С. В режиме черно-белой графики высоко-го разрешения случайную точку экрана соединим с точками границы экрана с заданным шагом. Полюбуйтесь, какие узоры получают-ся из-за дискретности раstra дисплея.

```

1 ПРОЦ МУАРЫ;
2 ВВОД ДАННЫХ: Ш;
3 РЖМ("ВГОП"); ЦВЕТ(1);
4 ""->К;
5 ПОКА НЕ К="К" : :
6 ДСЧ()*256->X; ДСЧ()*256->Y;
7 ДЛЯ И ОТ 0 ДО 255 ШАГ Ш : :
8 ЛИН(X,У,И,0); ЛИН(X,У,0,И);
9 ЛИН(X,У,И,255); ЛИН(X,У,255,И)
10 ВСЕ;
11 КЛАВ()->К
12 ВСЕ;
13 РЖМ("Т1ДП");
14 КНЦ;
10 REM МУАРЫ
20 INPUT S
30 MGR=1: COLOR=15
40 X=RND(1)*256: Y=RND(1)*256
50 FOR I=0 TO 255 STEP S
60 PLOT I,0; TO X,Y TO 0,I
70 PLOT I,255 TO X,Y TO 255,I
80 NEXT I
90 GET КВ
100 IF NOT КВ="К" THEN 40
110 ТЕХТ=2

```

К. После завершения рисунка в строке 11 (90) программа ждет нажатия любой клавиши, чтобы рисовать новый. Если нажата «К» — завершение с переключением в текстовый режим. ТЕХТ=2 — стандартная текстовая страница «нового» Бейсика, для «старого» — ТЕХТ=15.

Р. Попробуйте сделать не один, а два, три центра «излучения». Поэкспериментируйте в цветном режиме.

## 9. Комната

С. Нарисуем план квартиры или комнаты, расставим мебель. В Рапире это можно сде-

лать в режиме диалога, совмещенном с графикой. Из-за того, что нижнюю часть экрана при этом занимают 6 текстовых строк, вертикальные координаты должны принадлежать отрезку [24; 127] в графике среднего разрешения и [48; 255] — высокового.

```

#РЖМ("СГОК");
#ЦВЕТ(1);ОБЛ(63,63);           - пол
#ЦВЕТ(0);ПРЯМ(3,25,15,75);    - пианино
#ЦВЕТ(2);ПРЯМ(20,40,115,115); - ковер
#ЦВЕТ(3);
#ЛИН(60,110,90,110);ЛИН(90,110,90,60);
  ЛИН(90,60,60,60);ЛИН(60,60,60,110);
#ЦВЕТ(5);ОБЛ(80,80);           - стол
  
```

В Бейсике такого режима нет, нет и процедуры ПРЯМ, вместо которой написана соответствующая подпрограмма.

```

10 REM КОМНАТА
20 MGR=1
30 DATA 1,0,0,127,127 REM ПОЛ
40 DATA 0,3,25,15,75 REM ПИАНИНО
50 DATA 2,20,20,115,115 REM КОВЕР
60 DATA 3,60,60,90,110 REM СТОЛ
90 DATA -1
100 READ C
110 IF C<0 THEN 150
120 READ X1, Y1, X2, Y2
130 GOSUB 300
140 GOTO 90
150 COLOR= 3
160 PLOT 60,60 TO 60,110 TO 90,110 TO
      90,60 TO 60,60
200 END
300 REM ПРЯМОУГОЛЬНИК
310 COLOR=C
320 FOR X = X1 TO X2
330 PLOT X,Y1 TO X,Y2
340 NEXT X
350 RETURN
  
```

К. В строках 30—60 записаны цвета и координаты концов диагоналей прямоугольников, которые в строках 100—140 читаются и передаются подпрограмме 300, —1 — признак конца данных.

Р. Хорошо бы сделать программу, помогающую подобрать подходящий вариант расстановки мебели в квартире или комнате, подвигать мебель на экране. Конечно, это лучше сделать в черно-белой графике высокового разрешения. В начале программы должны вводиться размеры помещения и предметов. Выбранный вариант можно будет отпечатать, сохранить на диске.

## 10. Контур

С. Отрезков и прямоугольников для серьезных рисунков, для построения движущихся изображений, конечно, мало. Составим процедуру, соединяющую линиями несколько точек, которые зададим таблицей приращений координат каждой точки относительно предыдущей. Кроме этой таблицы входными данными для нашей процедуры будут координаты

первой точки контура («печки», от которой начнем «плясать») и коэффициенты увеличения по горизонтали и вертикали. В Бейсике придется также указать длину таблицы.

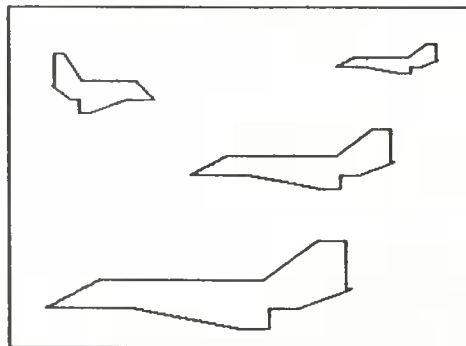
```

1 ПРОЦ САМОЛЕТЫ;
2 <10,7,30,0,10,10,5,0,0,-12,1,0,-10,
  -5,-5,0,0,-5,-5,0,-20,5,-16,0>->M1
3 РЖМ("ВГОП");ЦВЕТ(1);
4 КОНТУР(M,20,30,3,3);
5 КОНТУР(M,100,130,2,2);
6 КОНТУР(M,180,210,1,1);
7 КОНТУР(M,80,185,-1,2);
8 КНЦ;
9 ПРОЦ КОНТУР(K,X,Y,A,B);
10 ЕСЛИ K ВИДА (<) И #K>1 ТО
11 X->X1; Y->Y1;
12 ДЛЯ И ОТ 1 ДО #K-1 ШАГ 2 ::
13 X1+K[I]*A->X2; Y1+K[I+1]*B->Y2;
14 ЛИН(X1,Y1,X2,Y2);
15 X2->X1; Y2->Y1;
16 ВСЕ
17 ВСЕ;
18 КНЦ;
  
```

```

10 REM САМОЛЕТЫ
20 DATA 10,7,30,0,10,10,5,0,0,-12,1,0,
  -10,-5,-5,0,0,-5,-5,0,-20,5,-16,0
30 N = 12: DIM DX(N),DY(N)
40 FOR I=1 TO N: READ DX(I),DY(I)
50 NEXT I: HGR=1: COLOR=15
60 X= 20: Y=225: A= 3: B=3: GOSUB 200
70 X=100: Y=125: A= 2: B=2: GOSUB 200
80 X=180: Y= 45: A= 1: B=1: GOSUB 200
90 X= 80: Y= 70: A=-1: B=2: GOSUB 200
100 END
200 REM КОНТУР
210 IF N < 2 THEN RETURN
220 PLOT X,Y
230 XX = X: YY = Y
240 FOR I = 1 TO N
250 XX=XX+DX(I)*A: YY=YY-DY(I)*B
260 PLOT TO XX, YY
270 NEXT I
280 RETURN
  
```

К. В Рапире таблица приращений заносится в кортеж (строка 2), для цикла в строке 12 его длина определяется операцией  $\#$ . К. В Бейсике в строке 30 явно задается число точек контура N и определяются массивы для приращений. В строках 4—7 (60—90) идут обращения к процедуре рисования контура





с различными начальными координатами и коэффициентами увеличения. В строке 10 (210) делается проверка на непустоту таблицы приращений. Вертикальные приращения в Рапире (13) и Бейсике (250) берутся с противоположными знаками, потому что в Рапире ординаты отсчитываются снизу вверх, а в Бейсике — наоборот.

Р. Поэкспериментируйте с разными по величине и знаку коэффициентами увеличения. В цветной графике в число параметров процедуры КОНТУР войдет, конечно, номер цвета, который пригодится и в черно-белой графике для создания движущихся изображений путем закрашивания фигурки цветом фона и рисования ее в новом месте. Длину таблицы в Бейсике тоже можно вычислить, если, как в теме «Комната», ввести в качестве признака конца невозможное приращение, например:

```
10 REM ПРИМЕР
20 DATA 1,1,1,-1,-1,-1,-1,1,999
30 N=0
40 READ A
50 IF X<256 THEN N=N+1: GOTO 40
60 N=INT(N/2): DIM X(N),Y(N): RESTORE
70 FOR I=1 TO N: READ X(I),Y(I): NEXT
```

## 11. Прятки

С. Это двухмерное развитие темы УГАДАЙ. Машина загадывает координаты точки на цветном графическом дисплее, но не показывает ее. Человек ищет эту точку с помощью пультов, задающих координаты ищущей точки, которую машина показывает цветом, зависящим от расстояния между ищущей и спрятанной точками («горячо — холодно»). Когда точки сливаются, звучит сигнал, и тайник выделяется желтой каймой.

```
1 ПРОЦ ПРЯТКИ;
2 ЦЕЛЧ(ДСЧ()*128)->А; ЦЕЛЧ(ДСЧ()*128)
->Б;
3 РЖМ("СГОП"); 100->Р;
4 ПОКА Р>0 ::
5 ЦЕЛЧ(РУЧКА(1)/2)->Х;
ЦЕЛЧ(РУЧКА(2)/2)->У;
6 SQRT((А-Х)**2+(Б-У)**2)->Р;
7 ЦВЕТ(0); ТЧК(Х,У);
8 ЦВЕТ(Р/15+1); ТЧК(Х,У)
9 ВСЕ;
10 ?АЛО(7);
11 ЦВЕТ(3); ПРЯМ(А-1,Б-1,А+1,Б+1);
12 ЦВЕТ(1); ТЧК(А,Б);
13 КНЦ;

10 REM ПРЯТКИ
20 А=RND(1)*128 : В=RND(1)*128
30 MGR=1
40 Х=PDL(0)/2 : Y=PDL(1)/2
50 R%=SQR((Х-А)^2+(Y-В)^2)
60 COLOR=0
70 PLOT X,Y
80 COLOR=R%/15+1
90 PLOT X,Y
```

```
100 IF R% > 0 THEN 40
110 ? CHR$(135): COLOR=3
120 PLOT X-1,Y-1 TO X-1,Y+1 TO X+1,Y+1
130 PLOT X+1,Y+1 TO X+1,Y-1 TO X-1,Y-1
```

К. Поиск идет в бесконечном цикле, пока расстояние между точками больше нуля. Показания пультов, изменяющиеся от 0 до 255, приходится делить пополам, чтобы они оказались в диапазоне координат от 0 до 127. Чтобы отличить ищущую точку от оставленных ею следов того же цвета, в строках 7—8 (60—70) она сделана мерцающей.

Р. Исследуйте, как изменится характер поиска, если расстояние между точками делить не на 15, а на другое число — большее или меньшее. Попробуйте ввести звуковое сопровождение поиска — «минискатель» (см. тему «Звук»).

## 12. Перо

С. Если один пульт будет задавать абсциссу, а другой — ординату точки на экране, то, вращая ручки, можно рисовать. Нажимая кнопки на том или другом пульте, можно менять цвет и состояние пера: «поднято-опущено». Выход — по одновременному нажатию обеих кнопок. Еще интереснее рисовать вдвоем — это трудный тест на совместимость.

```
1 ПРОЦ ПЕРО;
2 РЖМ("СГОП"); 1->Ц; 0->С;
3 ЦВЕТ(Ц); 0->ХХ; 0->УУ;
4 ПОКА КНОПКА(1)="Д" ИЛИ
КНОПКА(2)="Д" ::
5 РУЧКА(1)/2->Х; РУЧКА(2)/2->У;
6 ЕСЛИ КНОПКА(1)="Н" ТО
7 Ц+1->Ц;
8 ЕСЛИ Ц=8 ТО 0->Ц ВСЕ;
9 ЦВЕТ(Ц); ПАУЗА(10);
10 ВСЕ;
11 ЕСЛИ КНОПКА(2)="Н" ТО
12 ABS(С-1)->С; ПАУЗА(10);
13 ВСЕ;
14 ЕСЛИ С=1 ТО
15 ЛИН(ХХ,УУ,Х,У)
16 ИНАЧЕ
17 ЦВЕТ(0); ТЧК(ХХ,УУ);
18 ЦВЕТ(Ц); ТЧК(Х,У)
19 ВСЕ;
20 Х->ХХ; У->УУ;
21 ВСЕ;
22 РЖМ("Т1ДП");
23 КНЦ;

10 REM ПЕРО
20 MGR=1 : C=1 : S=0 : COLOR=C
30 Х=PDL(0)/2 : Y=PDL(1)/2
40 А=PEEK(10061) : В=PEEK(10062)
50 IF А = 255 THEN 90
60 С = С + 1
70 IF С = 8 THEN С = 0
80 COLOR=C : GOSUB 200
90 IF В = 255 THEN 110
100 S = ABS(S - 1) : GOSUB 200
110 IF S=1 THEN PLOT ХХ,УУ: GOTO 140
```

```

120 COLOR=0: PLOT XX,YY
130 COLOR=C: PLOT X,Y
140 XX = X: YY = Y
150 IF A = 255 OR B = 255 THEN 30
160 TEXT= 2
170 END
200 REM ЗАДЕРЖКА
210 FOR I = 0 TO 100: NEXT : RETURN

```

К. Опрос кнопок производится с помощью функции КНОПКА (РЕЕК в строке 40). Для надежности срабатывания введена задержка функцией ПАУЗА (подпрограммой 200). При выходе происходит возврат к текстовому диалогу.

Р. Попробуйте убрать задержку, поменять ее величину. Если кроме пультов опрашивать еще и клавиши, можно построить мощный графический редактор типа ПГР (GREDIT).

### 13. Коды и символы

С. В машине каждой клавише соответствует код и какое-нибудь действие — отображение символа, писк динамика, очистка экрана, перевод строки, движение по экрану, изменение цвета и т. п. И если нажатие символьных клавиш мы можем распознавать из программы, сравнивая результат нажатия с символьной константой, то для работы с остальными клавишами надо знать их коды. И наоборот, полезно уметь через коды командовать машиной не с клавиатуры, а из программы. Таблицы кодов приводятся в документации, но их недолго определить и с помощью машины в режиме диалога или небольшой программкой.

```

#""->K;ПОКА НЕ K="K": КЛАВ()->K;?;
?K,КОД(K) ВСЕ;

```

```

10 REM КОДЫ СИМВОЛОВ
20 GET S#
30 ? : ?S#, ASC (S#)
40 IF S# <> "K" THEN 20

```

К. Приведенные выше строка и программа выдают действие и код нажатой клавиши, выход по «К». Указанные ниже строки выдают в цикле от 0 до 255 все коды и их действия. Быстрый вывод на экран можно приостанавливать клавишей ПРОБЕЛ.

```

#ДЛЯ А ОТ 0 ДО 255:?:?А,АЛФ(А) ВСЕ;
JFOR I=0 TO 255:?:?А,CHR#(I): NEXT

```

Р. Напечатайте себе памятку — таблицу кодов и символов.

### 14. Бегущая точка

С. Заставим бегать по экрану графики низкого разрешения 64×64 цветную точку, меняющую направление движения по коман-

дам клавиш со стрелками и останавливаемую клавишей РЕД.

```

1 ПРОЦ БЕТОЧКА;
2 РЖМ("Н10П"); ""->НАЖ;
3 31->X; 31->Y; 0->ШX; 0->ШY;
4 ПОКА НЕ НАЖ="K" ::
5 НАЖАТО()->НАЖ;
6 ВЫБОР НАЖ ИЗ
7 АЛФ(136):-1->ШX; 0->ШY !
8 АЛФ(149): 1->ШX; 0->ШY !
9 АЛФ(153): 0->ШX; 1->ШY !
10 АЛФ(154): 0->ШX;-1->ШY !
11 АЛФ(155): 0->ШX; 0->ШY
12 ВСЕ;
13 ЦВЕТ(0); ТЧК(X,Y);
14 X+ШX->X; Y+ШY->Y;
15 ЕСЛИ X<0 ИЛИ X>63 TO X-ШX->X ВСЕ;
16 ЕСЛИ Y<0 ИЛИ Y>63 TO Y-ШY->Y ВСЕ;
17 ЦВЕТ(1); ТЧК(X,Y); ПАУЗА(2);
18 ВСЕ;
19 РЖМ("Т1ДП");
20 КНЦ;

```

```

10 REM БЕГУЩАЯ ТОЧКА
20 GR=2
30 X=31 : Y=31 : SX=0 : SY=0
40 K=РЕЕК(КС000) : РОКЕ КС010,0
50 IF K=ASC("K") THEN TEXT=2: END
60 IF K=136 THEN SX=-1: SY=0: GOTO 140
70 IF K=149 THEN SX= 1: SY=0: GOTO 140
80 IF K<153 OR K>155 THEN 140
90 ON K-152 GOTO 100,110,120
100 SY =-1: GOTO 130
110 SY = 1: GOTO 130
120 SY = 0
130 SX = 0
140 COLOR=0: PLOT X,Y
150 X = X + SX : Y = Y + SY
160 IF X < 0 OR X > 63 THEN X = X - SX
170 IF Y < 0 OR Y > 63 THEN Y = Y - SY
180 COLOR=1 : PLOT X,Y
190 GOTO 40

```

К. В строке 3 (30) задаются исходные значения координат точки и горизонтального, и вертикального шага. В строке 5 (40) опрашивается клавиатура, и код нажатой клавиши (или 0, если ничего не нажимали) в строках 6—12 (50—90) сравнивается с кодами управляющих клавиш; в случае совпадения задаются новые значения шагов. Затем старая точка стирается и ставится новая. Условия в строках 15—16 (160—170) останавливают точку на границах экрана. Выход — по «К».

Р. Конечно, можно гонять не точку, а более интересную фигурку, и не по пустому полю, а по какому-нибудь лабиринту. Введя в игру дополнительные клавиши, можно заставить точку (и не одну) бегать по диагонали, менять цвет, оставлять за собой след. Вместо остановки на границе можно сделать переход на противоположную сторону экрана.

### 15. График 1

С. Научим машину благородному искус-

ству графического изображения зависимости между числами. Начнем с построения графика функции. Необходимо будет задать формулу вычисления функции по аргументу, затем диапазоны изменения значений аргумента и функции. Математические абсциссу и ординату для размещения на экране нам придется пересчитывать в экранные координаты и наоборот. Обязательно нарисуем оси — через 0 или крайние значения. В цикле слева направо по экранной абсциссе будем вычислять соответствующий ей математический аргумент, по аргументу — функцию, а по значению функции — экранную ординату и, если эта ордината не оказалась за пределами экрана, ставить точку.

```

1 ПРОЦ ГРАФИК(Ф,Х1,Х2,У1,У2);
2 ОСИВ(Х1,Х2,У1,У2);
3 ДЛЯ А ОТ А1 ДО А2 ::
4   Ф(Х1+(А-А1)*МХ)->У;
5   В1+(У-У1)/МУ->В;
6   ЕСЛИ У>=У1 И У<=У2 ТО ТЧК(А,В) ВСЕ
7   ВСЕ;
8   НАДПИСЬ( );
9 КНЦ;

```

```

10 ПРОЦ ОСИВ(Х1,Х2,У1,У2);
11 ЕСЛИ Х2<=Х1 ИЛИ У2<=У1 ТО СТОП ВСЕ;
12 Ø->ХØ; Ø->УØ;
13 ЕСЛИ Х1>=Ø ТО Х1->ХØ
14 ИНАЧЕ ЕСЛИ Х2<=Ø ТО Х2->ХØ ВСЕ
15 ВСЕ;
16 ЕСЛИ У1>=Ø ТО У1->УØ
17 ИНАЧЕ ЕСЛИ У2<=Ø ТО У2->УØ ВСЕ
18 ВСЕ;
19 З->А1; 252->А2; З->В1; 252->В2;
20 (Х2-Х1)/(А2-А1)->МХ;
21 (У2-У1)/(В2-В1)->МУ;
22 А1+(ХØ-Х1)/МХ->АØ;
23 В1+(УØ-У1)/МУ->ВØ;
24 РЖМ("ВГОП"); ЦВЕТ(1);
25 ЛИН(А1,ВØ,А2,ВØ); ЛИН(АØ,В1,АØ,В2);
26 ЛИН(А2-5,ВØ-2,А2,ВØ);
   ЛИН(А2-5,ВØ+2,А2,ВØ);
27 ЛИН(АØ-2,В2-5,АØ,В2);
   ЛИН(АØ+2,В2-5,АØ,В2);
28 КНЦ;

```

```

29 ФУНК Ф(Т); Ø->Р;
30 ЕСЛИ Т/=-2 ТО Т+1/(Т+2)**2-1->Р ВСЕ;
31 РЕЗ:Р;
32 КНЦ;

```

```

33 ПРОЦ НАДПИСЬ;
34 ТКС(96,30,8,0,1,0,1,2,1,2,1,0,
   "у=x+1/(x+2)^2-1");
35 КНЦ;

```

```

10 REM ГРАФИК
20 HOME : ?" СТРОИМ ГРАФИК ФУНКЦИИ : "
30 LIST 1000,1999: ?
40 ?" ЗАДАЙТЕ ГРАНИЦЫ АРГУМЕНТА И
   ФУНКЦИИ"
50 ? : INPUT " XMIN, XMAX "; X1, X2
60 IF X1 >= X2 THEN Ø
70 ? : INPUT " YMIN, YMAX "; Y1, Y2
80 IF Y1 >= Y2 THEN Ø
300 REM ОСИ HGR

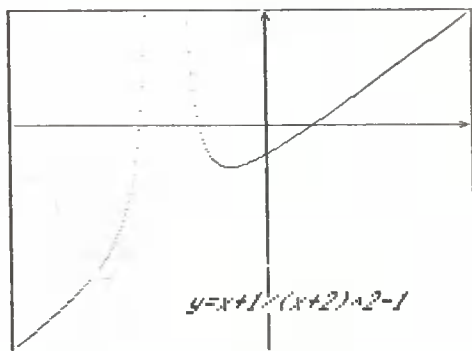
```

```

310 ХØ = Ø: УØ = Ø
320 IF Х1 > Ø THEN ХØ = Х1: GOTO 340
330 IF Х2 < Ø THEN ХØ = Х2
340 IF Y1 > Ø THEN УØ = Y1: GOTO 360
350 IF Y2 < Ø THEN УØ = Y2
360 А1 = 3: А2 = 252: В1 = 252: В2 = 3
370 МХ = (Х2-Х1)/(А2-А1)
380 МУ = (У2-У1)/(В2-В1)
390 АØ = А1 + (ХØ-Х1)/МХ
400 ВØ = В1 + (УØ-У1)/МУ
410 HGR=1: COLOR=15
420 PLOT А1,ВØ TO А2,ВØ: PLOT АØ,В1 TO
   АØ,В2
430 PLOT А2-5,ВØ-2 TO А2,ВØ TO
   А2-5,ВØ+2
440 PLOT АØ-2,В2+5 TO АØ,В2 TO
   АØ+2,В2+5
500 REM ТОЧКИ ГРАФИКА
510 FOR А = А1 TO А2
520 Х = Х1+(А-А1)*МХ : GOSUB 1000
530 В = В1+(У-У1)/МУ
540 IF В>=В2 AND В<=В1 THEN PLOT А,В
550 NEXT А
560 END
1000 REM ВЫЧИСЛЕНИЕ ФУНКЦИИ
1010 IF Х<>-2 THEN У=Х+1/(Х+2)^2-1
1020 RETURN

```

К. На иллюстрации видно, что должно получиться в Рапире после обращения #ГРАФИК (ФФ,—5,4,—6,3);, содержащего и имя функции, и необходимые диапазоны.



Текст надписи малыми буквами набран в «Редакторе отладчика». Для построения графика другой функции надо будет написать другой блок ФУНК с другим именем, изменить текст в НАДПИСЬ и обратиться к ГРАФИК с соответствующими именем и диапазонами. В Бейсике труднее совмещать текст с графиком, но зато Бейсик-программа может распечатать саму себя, в частности текст подпрограммы вычисления функции, а диапазоны можно ввести и в диалоге, заодно проверив их на здравый смысл. В отдельный блок ОСИВ (300—400) выделено вычисление масштабов и построение осей, все это еще пригодится.

Р. Конечно, осям нужна разметка и оцифровка. Для рисования цветных графиков можно переделать ОСИВ в ОСИС с соот-

ветствующими изменениями границ экран-ных координат А и В. Для ускорения рисо-вания попробуйте идти по абсциссе не подряд, а с некоторым шагом, соединяя точ-ки линиями.

## 16. График 2

С. Если в исследуемом диапазоне аргу-мента нет точек, приближаясь к которым функция стремится в бесконечность, то поче-му бы не поручить машине самой найти ее минимум и максимум, пройдя диапазон аргумента с некоторым шагом. Чтобы не де-лать повторных вычислений, значения функции можно запомнить в соответствующем кортеже (массиве). Иногда функцию задают в виде таблицы, или таблицы по-лучаются в результате расчетов, причем шаг таблицы по аргументу может быть неравно-мерным. В нашей программе сначала будут созданы таблицы значений аргумента и функции, найдены минимумы и максимумы, а затем по этим данным будет построен график.

```

1 ПРОЦ ГРАФ(Ф,Е1,Е2);
2 <->Ц; <->Д; (Е2-Е1)/50->ДЕ;
3 ДЛЯ Е ОТ Е1 ДО Е2 ШАГ ДЕ ::
4 Ц+(Е->Ц; Д+(Ф(Е))->Д
5 ВСЕ;
6 ГРАФИН(Ц,Д);
7 КНЦ;

8 ФУНК F(X); РЕЗ:Х*Х-2*Х-3; КНЦ;

9 ПРОЦ НАДПИСЬ;
10 ТКС(100,200,8,0,1,0,1,2,1,2,1,0,
"у=x^2-2x-3");
11 КНЦ;

12 ПРОЦ ГРАФИН(Х,У);
13 ЕСЛИ #Х>#У ТО СТОП ВСЕ;
14 Х[1]->Х1; Х1->Х2; У[1]->У1; У1->У2;
15 1->И1; 1->И2;
16 ДЛЯ И ОТ 2 ДО #Х ::
17 ЕСЛИ Х[И1]>Х1 ТО Х[И1]->Х1;
18 ИНАЧЕ ЕСЛИ Х[И1]>Х2 ТО Х[И1]->Х2 ВСЕ
19 ВСЕ;
20 ЕСЛИ У[И1]>У1 ТО У[И1]->У1; И->И1
21 ИНАЧЕ ЕСЛИ У[И1]>У2 ТО У[И1]->У2;
И->И2 ВСЕ

22 ВСЕ
23 ВСЕ;
24 ?;"НА ОТРЕЗКЕ [" ,Х1," ; ",Х2," ]";
25 ?;"УМИН=" ,У1," ПРИ Х=" ,Х[И1];
26 ?;"УМАХ=" ,У2," ПРИ Х=" ,Х[И2];
27 КЛАВ()->К;
28 ОСИВ(Х1,Х2,У1,У2);
29 ДЛЯ И ОТ 2 ДО #Х ::
30 А1+(Х[И-1]-Х1)/МХ->АН;
31 В1+(У[И-1]-У1)/МУ->ВН;
32 А1+(Х[И]-Х1)/МХ->АК;
33 В1+(У[И]-У1)/МУ->ВК;
34 ЛИН(АН,ВН,АК,ВК)
35 ВСЕ;
36 НАДПИСЬ();

```

```

37 ПОКА НЕ К="К" ::
38 КЛАВ()->К; РЖМ("ВГП");
39 КЛАВ()->К; РЖМ("ТІП");
40 ВСЕ;
41 КНЦ;

```

```

10 REM ГРАФИН
20 HOME : ?" СТРОИМ ГРАФИК ФУНКЦИИ : "
30 LIST 40: ?
40 DEF FN F(X) = Х*Х - 2*Х - 3
50 ?" ЗАДАЙТЕ ГРАНИЦЫ АРГУМЕНТА"
60 ? : INPUT " XMIN, XMAX "; X1, X2
70 IF X1 >= X2 THEN 50
80 N=50: DIM X(N),Y(N): DX=(X2-X1)/N
90 X(0) = X1: Y(0)=FN F(X1): Y1=Y(0)
100 Y2 = Y1: I1 = 0: I2 = 0
110 FOR I = 1 TO N: X(I)=X(I-1)+DX
120 Y(I) = FN F(X(I))
130 IF Y(I) < Y1 THEN Y1=Y(I): I1=I:
GOTO 150
140 IF Y(I) > Y2 THEN Y2=Y(I): I2=I
150 NEXT I
160 ? : ?"УМИН="; Y1;" ПРИ Х=";X( I1)
170 ? : ?"УМАХ="; Y2;" ПРИ=";X( I2)
180 GET К;
300 REM ОСИ HGR

```

. сн тему ГРАФИК. 1

```

500 REM ЛИНИИ ГРАФИКА
510 А = А1 + (Х(0)-Х1)/МХ
520 В = В1 + (У(0)-У1)/МУ
530 PLOT А,В
540 FOR I = 1 TO N
550 PLOT TO А1+(Х(I)-Х1)/МХ,
В1+(У(I)+У1)/МУ
560 NEXT I
570 GET К; TEXT=2
580 IF К = "К" THEN END
590 GET К; POKE ПС713,0
600 GOTO 570

```

К. В этой Бейсик-программе формула не заведена в подпрограмму, вызываемую по номеру строки, а определена как функция с именем F (строка 40). Диапазон изменения аргумента делится на 50 частей, в 51 точке вычисляются значения функции, в строках 13—23 (90—150) из них выбираются наименьшее и наибольшее, результат выводится на экран. По нажатию какой-нибудь клавиши (строка 27/180) рисуются оси и все точки соединяются линиями, после этого любая клавиша (кроме «К» — выход) переключает из графического режима в текстовый и обратно.

Р. Поэкспериментируйте с числом отрезков, на которые разбивается диапазон аргумента. Постройте изображение неоднозначной функции, например спирали.

## 17. Словарь

С. Составим обучающую программу, содержащую некоторую часть англо-русского словаря и позволяющую работать в двух ре-

жимак: опрос, т. е. на заданные программой русские слова надо ввести соответствующие английские, после трех ошибок в переводе слова программа подсказывает, и перевод, т. е. для введенных английских слов программа находит русские, а если не находит, то извиняется.

```

1 ПРОЦ СЛОВАРЬ;
2 <"IF","ЕСЛИ","ТНЕН","ТОГДА,ТО","FOR"
  ,"ДЛЯ","END","КОНЕЦ","СТЕР","ШАГ",
  ,"PRINT","ПЕЧАТАТЬ","GET","ПОЛУЧИТЬ",
  ,"READ","ЧИТАТЬ","PLOT","ЧЕРТИТЬ",
  ,"INPUT","ВВОД","ТО","ДО,К","COLOR",
  ,"ЦВЕТ","RETURN","ВОЗВРАТ","DATA",
  ,"ДАННЫЕ">->С;
3 ПОКА I=1 ::
4   ВЫВОД БПС:"O -ОПРОС, П -ПЕРЕВОД,
   К -КОНЕЦ";?;
5   ВЫБОР КЛАВ() ИЗ
6     "O": ОПРОС() !
7     "П": ПЕРЕВОД() !
8     "К": ВЫХОД
9   ВСЕ
10 ВСЕ;
11 КНЦ;

12 ПРОЦ ОПРОС;
13 ДЛЯ И ОТ 2 ДО #С ШАГ 2 ::
14   ВЫВОД БПС:С[И], " - "; Ø->Н;
15   ПОКА Н<3 :: ВВОД:ОТВЕТ;
16   ЕСЛИ ОТВЕТ=С[И-1] ТО
17     ?" ПРАВИЛЬНО"; 3->Н
18   ИНАЧЕ ?" НЕПРАВИЛЬНО";Н+1->Н;
19   ЕСЛИ Н=3 ТО ?" ",С[И-1] ВСЕ
20   ВСЕ
21 ВСЕ;
22 КНЦ;

23 ПРОЦ ПЕРЕВОД;
24 ПОКА I=1 ::
25   ?"ВВЕДИТЕ ИНТЕРЕСУЮЩЕЕ ВАС";
26   ?" АНГЛИЙСКОЕ СЛОВО ИЛИ К"?;?;
27   ВВОД:ВОПРОС;
28   ЕСЛИ ВОПРОС="К" ТО ВЫХОД
29   ИНАЧЕ -1->И;
30   ПОКА И<#С-2 :: И+2->И;
31   ЕСЛИ ВОПРОС=С[И] ТО
32     ?" ",С[И+1]; #С+2->И ВСЕ
33   ВСЕ;
34   ЕСЛИ И=#С-1 ТО ?"EXCUSE ME,
   I DON'T KNOW";?; ВСЕ
35 ВСЕ
36 КНЦ;

```

```

10 REM СЛОВАРЬ
20 DATA IF,ЕСЛИ,ТНЕН,"ТОГДА,ТО",FOR
25 DATA ДЛЯ,END,КОНЕЦ,СТЕР,ШАГ,PRINT
30 DATA ПЕЧАТАТЬ,GET,ПОЛУЧИТЬ,READ
40 DATA ЧИТАТЬ,PLOT,ЧЕРТИТЬ,INPUT,ВВОД
45 DATA ТО,"ДО,К",COLOR,ЦВЕТ,RETURN
50 DATA ВОЗВРАТ,DATA,ДАННЫЕ
60 N = 14: DIM ЕЯ(N),RЯ(N)
70 FOR I=1 TO N:READ ЕЯ(I),RЯ(I):NEXT I
80 ?"1 -ОПРОС, 2 -ПЕРЕВОД, Ø -КОНЕЦ";
90 GET КЯ: HOME
100 ON VAL(КЯ)+1 GOSUB 120,200,300
110 GOTO 80

```

```

120 END
200 REM ОПРОС
210 FOR I=1 TO N: ?RЯ(I);" - "; К=Ø
220 INPUT RЯ
230 IF RЯ=ЕЯ(I) THEN ?" ПРАВИЛЬНО":
   GOTO 260
240 ?" НЕПРАВИЛЬНО": К=К+1:
   IF К < 3 THEN 220
250 ?" ";ЕЯ(I)
260 NEXT I
270 RETURN
300 REM ПЕРЕВОД
310 ?"ВВЕДИТЕ ИНТЕРЕСУЮЩЕЕ ВАС"
320 ?" АНГЛИЙСКОЕ СЛОВО ИЛИ К"
330 ?;INPUT QЯ: IF QЯ = "К" THEN RETURN
340 FOR I = 1 TO N
350 IF QЯ = ЕЯ(I) THEN ?" ";RЯ(I):
   GOTO 310
360 NEXT I
370 ?"EXCUSE ME, I DON'T KNOW": ?;
   GOTO 310

```

К. В Бейсике русские переводы, содержащие запятую, разделяющую варианты, заключены в кавычки, без которых эти варианты оказались бы в разных элементах массива и все запуталось бы. В Бейсике английские и русские слова помещены в два массива, перебираемые в циклах подряд, а в Рапире все слова записаны с чередованием в один кортеж, который просматривается в циклах с шагом 2. В Бейсик-программе при выборе режима ОПРОС/ПЕРЕВОД в строке 90 вводится символ цифры 1 или 2, который в 100-й строке с помощью функции VAL превращается в число для вызова одной из подпрограмм.

Р. Наверное, программа опроса должна подсчитывать количество или процент правильных и неправильных ответов и по какому-то критерию оценивать ответ, может быть, направлять опрос тем или другим путем. Серьезной программе нужно будет большее количество слов, хранимых в отдельных файлах (для разных языков), где слова будут упорядочены по алфавиту, что позволит программе перевода быстрее ориентироваться среди них (например, той же дихотомией: делим словарь пополам и смотрим, где интересующая нас буква — до или после и т. д.). При опросе также будут перебираться не все слова, а только некоторое количество слов, выбираемых из словаря случайно или по какому-нибудь принципу. Сделайте специальную программу, упорядочивающую набор слов по алфавиту — русскому или латинскому.

## 18. Звук

С. Игровую программу может приятно оживить удачно подобранное звуковое сопровождение. В Рапире для этого всегда под рукой встроенная процедура ЗВУК (Д, П) — 0<Д, П<256; чем больше П, тем ниже звук,

длительность звука зависит от Д и П. В Бейсике для этого заносят в соответствующее место памяти небольшую подпрограмму в кодах, к которой обращаются после засылки по соответствующим адресам параметров, аналогичных П и Д.

```

1 ПРОЦ З;
2 ПОВТОР З :: ЗВУК(80,166) ВСЕ;
3 ЗВУК(255,210);
4 ПОВТОР З :: ЗВУК(70,188) ВСЕ;
5 ЗВУК(255,224);
6 КНЦ;

7 ПРОЦ СИРЕНА(Д1,П1,Д2,П2,К);
8 ПОВТОР К ::
9 ЗВУК(Д1,П1); ЗВУК(Д2,П2)
10 ВСЕ;
11 КНЦ;

100 REM ЗВУК
200 DATA 160,0,174,0,3,173,48,192,136,
      208,5,206,1,3,240,5,202,208,
      245,240,237,96
300 FOR A=770 TO 791: READ C: POKE A,C:
      NEXT A
400 DATA 166,60,166,60,166,60,210,255,
      187,60,187,60,187,60,222,255
500 FOR I = 1 TO 8
600 READ F,L: POKE 768,F: POKE 769,L:
      CALL 770
700 NEXT I
800 GET К;
900 FOR I = 1 TO 6
1000 POKE 768,140: POKE 769,50: CALL 770
1100 POKE 768,70: POKE 769,100: CALL 770
1200 NEXT I

```

К. Строки 20—30 — ввод кодовой программы. Процедура З (строки 40—70) — воспроизведение знакомой энергичной музыкальной фразы. После вызова СИРЕНА (100, 150, 200, 70, 6) (в Бейсике после нажатия клавиши) звучит сирена.

Р. Полезно исследовать смысл, влияние и взаимосвязь параметров П и Д, разобраться в исходном ассемблерном тексте кодовой программы. На этой основе можно сделать программу, превращающую клавиатуру машины в музыкальный инструмент. Другие подобные программы могли бы создавать различные шумовые эффекты: рев мотора, стук сердца, капель и даже подобию речи.

## 19. Музыкальная библиотека

С. Чередующиеся в определенном порядке звуки определенной частоты и длительности — это уже мелодия. В Рапире для этого на системном диске есть К-файл МУЗЫКА. КОД и процедура НОТА с параметрами КАКАЯ и ДЛИНА. КАКАЯ — это номер ноты от 1 до 47 (больше — выше), 0 — пауза. ДЛИНА от частоты не зависит и мо-

жет быть от 1 до 255, для целой ноты можно принять 64.

В Бейсике придется поступить так же, как в теме ЗВУК, но кодовая программа будет подлиннее и позволит получать более длинные звуки. В связи с тем что отношение частот соседних нот (полутон) равно корню 12-й степени из 2, используемые для получения нефальшивых звуков числа должны принадлежать соответствующей геометрической прогрессии.

Выбрав одну из нот в качестве опорной и отсчитывая от нее полтона вверх и вниз, можно закодировать мелодию в виде последовательности номеров нот (членов прогрессии). При этом для каждой ноты нужно задать длительность, но в связи с тем, что ноты одинаковой длины зачастую соседствуют, можно сэкономить, кодируя длительность отрицательным числом и считая, что она относится ко всем последующим нотам до новой длительности. Если ко всем номерам добавить одно и то же число, получится та же мелодия, но в другой тональности, повыше или пониже.

Несколько мелодий объединим в музыкальную библиотеку, выдающую по номеру одну из мелодий или их каталог для выбора.

```

1 ПРОЦ БИМ; (* БИБЛИОТЕКА МЕЛОДИЙ *)
2 ЕСЛИ НЕ ПРИЗНАК="ЗАГРУЖЕНО" ТО
3 ВВОД ИЗ ДЗУ:НОТА;
4 ПОДГОТОВИТЬ_НОТУ();
5 "ЗАГРУЖЕНО"->ПРИЗНАК
6 ВСЕ;
7 <<"ГАММА",1,3,5,6,8,10,12,13,12,10,8,
      6,5,3,1>>->Б;
8 <-8,8,-16,5,5>->К1; <8,6,5,3,-4,1>
      ->К2;
9 Б+<<"МАЛЕНЬКАЯ ЕЛОЧКА">+К1+К1+К2+
      <-8,10,-16,13,10>+К1+К2>->Б;
10 <-16,7,2,7,2,7,6,6,0,6,2,6,2,6>->К1;
11 <7,9,-32,9,9,-16,9,9,10,-32,10,10,
      -16,10,10,10,9,7,6>->К2;
12 Б+<<"КУЗНЕЧИК">+К1+<7,7,0>+К1+<-8,7,
      -16,0>+К2+<7,7,0>+К2+<-8,7>->Б;
13 Б+<<"СТЕПЬ",-8,1,-16,1,-8,3,-16,1,
      -3,10,-8,10,-16,10,-8,11,-16,10,-3,
      3,-8,3,-16,5,8,6,3,-3,1,-16,1,5,8,
      -8,11,-16,5,-3,6>->Б;
14 <-8,1,9,9,9,9,-4,9,-8,8,8,11,11,9,8,
      -4,8,-8,6,6,6,6>->К1;
15 Б+<<"ПРЕКРАСНОЕ ДАЛЕКО">+К1+<6,6,-5,
      4,-16,9,13,13,-3,11,-16,13,14,-3,
      13>+К1+<4,2,-16,1,6,-8,9,0,1,1,5,
      8,5,-4,6>->Б;
16 Б+<<"ПОДМОСКОВНЫЕ ВЕЧЕРА",-8,1,4,8,
      4,-4,6,-8,4,3,-4,8,6,-2,1,-8,4,8,
      11,11,-4,13,-8,11,9,-2,8,-4,10,12,
      -8,15,13,-3,8,-8,3,1,8,6,-2,9,-8,
      11,9,-4,8,-8,6,4,-4,8,6,-2,1>->Б;
17 <-32,7,6,6,-16,6>->К1;
      <-32,6,4,-16,4>->К2;
18 Б+<<"МОЦАРТ">+К1+К1+К1+<-8,14,-32,
      14,12,-16,11,-32,11,9,-16,7,-32,
      7,6,-16,4,-8,4>+К2+К2+К2+<-8,12,

```

```

-32,12,11,-16,10,-32,10,7,-16,6,
-32,6,4,-16,2,-8,2>>->B;
19 Ø->И; Ø->СДВИГ; 8->Д;

20 ПОКА И>=0 ::
21 ЕСЛИ И<1 ИЛИ И>#Б ТО
22 ДЛЯ Й ОТ 1 ДО #Б ::
    ?Й," ",Б[Й,1] ВСЕ
23 ИНАЧЕ
24 ?Б[И,1],", СДВИГ=",СДВИГ;?;
25 ВЫВОД БПС:
    "ИЗМЕНИМ ТОНАЛЬНОСТЬ (Д/Н) ?";
26 КЛАВ()->ОТВЕТ;?;Б[И,2:#Б[И1]->И;
27 ЕСЛИ ОТВЕТ="Д" ТО
28 48->МИН; Ø->МАК;
29 ДЛЯ Т ИЗ М ::
30 ЕСЛИ Т>Ø ТО
31 ЕСЛИ Т<МИН ТО Т->МИН ВСЕ;
32 ЕСЛИ Т>МАК ТО Т->МАК ВСЕ
33 ВСЕ
34 ВСЕ;
35 ?"ВЫБЕРИТЕ СДВИГ ОТ ",1-МИН,
    " ДО ",48-МАК;

36 ВВОД ДАННЫХ:СДВИГ;
37 ВСЕ;
38 МЕЛОДИЯ(М,СДВИГ)
39 ВСЕ;
40 ?;?" Ø - КАТАЛОГ МЕЛОДИЙ";
41 ?;?"1 -",#Б," - ИСПОЛНЕНИЕ";
42 ?;?" -1 - ВЫХОД";?;
43 ВВОД ДАННЫХ:И;
44 ВСЕ;
45 КНЦ;
46 ПРОЦ МЕЛОДИЯ(К,С);
47 ИМЕНА: Т,Д; 8->Д;
48 ДЛЯ Т ИЗ К ::
49 ВЫБОР ИЗ
50 Т<Ø: ЦЕЛЧ((-48/Т)->Д !
51 Т=Ø: НОТА(Т,Д) !
52 Т>Ø: НОТА(Т+С,Д)
53 ВСЕ
54 ВСЕ;
55 КНЦ;

100 REM БИБЛИОТЕКА МЕЛОДИЙ
200 GOSUB 1000
300 FOR A=1 TO N: ?A;" - ";NAME$(A):
    NEXT A
400 ?; INPUT K
500 IF K<Ø THEN END
600 IF K=Ø OR K>N THEN 300
700 ?; ?NAME$(K),"СДВИГ=";S: ?
800 ?"ИЗМЕНИМ ТОНАЛЬНОСТЬ (Д/Н) ?":
    GET R#
900 IF R#="D" OR R#="Д" THEN
    INPUT"СДВИГ ";S
1000 READ C
1100 IF C-1000<K THEN 1000
1200 GOSUB 900: RESTORE
1300 FOR A=770 TO 819+N: READ R#; NEXT A
1400 ?; ?" Ø - КАТАЛОГ МЕЛОДИЙ"
1500 ?; ?"1 - ";N;" - ИСПОЛНЕНИЕ"
1600 ?; ?" -1 - ВЫХОД"
1700 GOTO 400
900 REM *** МЕЛОДИЯ ***
910 READ C
920 IF C>255 THEN RETURN
930 IF C<Ø THEN POKE 769,-64/C:GOTO 910
940 POKE 768,N%(C+S): CALL 770:GOTO 910
950 RETURN
1000 REM *** НАЧАЛЬНЫЕ УСТАНОВКИ ***
1010 DATA 173,1,3,72,10,10,10,10,41,

```

```

240,141,64,3,104,74,74,74,41,
15,141,65,3,160,0,174,0,3,173,48,
192,136,208,10,206,64,3,208,5,206,
65,3,48,5,202,208,240,240,232,96
1020 FOR A=770 TO 819: READ C:
    POKE A,C: NEXT A
1030 DIM N%(37):A=256:P=2^(1/12):
    N%(1)=0
1040 FOR C=2 TO 37: A=A/P
1050 N%(C)=INT(A+0.5): NEXT C
1060 DATA ГАММА,МАЛЕНЬКАЯ ЕЛОЧКА,МОЦАРТ
1070 N=3: DIM NAME$(N)
1080 FOR A=1 TO N: READ NAME$(A):NEXT A
1090 RETURN
2000 REM ***** НОТЫ
2010 REM *** ГАММА ***
2011 DATA 1001,-8,1,3,5,6,8,10,12,13,
    13,12,10,8,6,5,3,1,256
2020 REM *** МАЛЕНЬКАЯ ЕЛОЧКА ***
2021 DATA 1002,-8,8,-16,5,5,-8,8,-16,5,
    5,8,6,5,3,-5,1,-8,10,-16,13,10,-8,
    8,-16,5,5,8,6,5,3,-5,1,256
2030 REM *** МОЦАРТ ***
2031 DATA 1003,-32,7,6,-16,6,-32,7,6,
    -16,6,-32,7,6,-16,6,-8,14,-32,14,
    13,-16,11,-32,11,9,-16,7,-32,7,6,
    -16,4,-8,4,-32,6,4,-16,4,-32,6,4,
    -16,4,-32,6,4,-16,4,-8,13,-32,13,
    11,-16,10,-32,10,7,-16,6,-32,6,4,
    -16,2,-8,2,256

```

61

К. В Рапира-программе первый раз загружается с диска файл НОТА с процедурой ПОДГОТОВИТЬ НОТУ, загружающей, в свою очередь, К-файл МУЗЫКА.КОД, и задается ПРИЗНАК загруженности этих файлов, по которому при последующих вызовах БИМа обращения к диску обходятся. При подборе тональности в строках 28—36 определяются минимальный и максимальный номера нот, дабы при сдвиге не случилось выхода за пределы диапазона 1—47. В процедуре МЕЛОДИЯ в цикле воспроизведения коды разделяются на длительности (меньшие нуля), ноты и паузы (равные нулю), к последним сдвиг не применяется. К этой процедуре можно обращаться и в диалоге, задавая кортеж мелодии в заголовке, подбирать ее на слух. Библиотеку можно расширять, наращивать кортеж Б новыми кортежами, начинающимися с текста-названия. Обратите внимание, как с помощью рабочих кортежей K1 и K2 вводятся повторяющиеся фрагменты.

В Бейсик-программе сначала исполняется подпрограмма 1000, где в память машины заносится кодовая программа управления динамиком, вычисляются и запоминаются в массиве N % 37 членов геометрической прогрессии — три октавы. Начало прогрессии — 256 (0) — самый низкий звук, но можно выбрать и другое начало, например 249 — на четверть тона выше. Обработки паузы нет, ее надо организовать специально, видимо, циклом. В масси-

ве NAME□ запоминаются названия мелодий для каталога. Поиск мелодии (100—110) в списке данных ведется по числу (1000+номер в каталоге), с которого она должна начинаться. После воспроизведения (в подпрограмме 900) по команде PESTORE происходит возврат на начало списка и «пустое» считывание всех начальных данных (как текстовых) для подхода к первой мелодии.

Экспериментируя со сдвигом тональности, вы можете убедиться, что слишком высокие ноты звучат хуже, слышится фальшь, вызванная ошибками округления до целого. Для сравнения распечатайте геометрическую прогрессию с округлением и без него.

Р. Можно изменять не только тональность, но и темп исполнения (adagio, allegro и др.), вводя множитель у длительности. Вставляя между всеми нотами небольшие паузы, можно попробовать получить staccato. Вообще, полезно было бы облегчить процесс кодирования мелодий с использованием традиционных названий нот (до, до<sup>♯</sup>, ре...) и других музыкальных стандартов (мажор, минор и проч.). И все это в графике — нотный стан, скрипичный ключ, нотные знаки. При этом необходимо помнить, что быстрое действие машины невелико и задержки при воспроизведении мелодии не доставят удовольствия слушателям.

## 20. Двоичные числа

С. В машине вся информация хранится в двоичном виде. Значением того или иного двоичного разряда (0 или 1) вместе с соседними определяются цвет, символ, изображение в графике и даже оператор в Бейсике. Научимся переводить десятичные числа в двоичную систему счисления и обратно. Двоичные числа будем представлять в виде текстов, состоящих из символов 0 и 1.

```

1 ФУНК BYNDEC(DEC); (* 10 -> 2 *)
2 ИМЕНА: BYN,A,B;
3 ЕСЛИ DEC ВИДА 0 ТО
4   ""->BYN; DEC->A;
5   ПОКА A>0 :: A//2->B;
6   ЕСЛИ A>B ТО "1"+BYN->BYN
7   ИНАЧЕ "0"+BYN->BYN
8   ВСЕ;
9   B->A
10  ВСЕ
11 ИНАЧЕ
12  ?"ОШИБКА: BYNDEC(",DEC,")"; ВЫХОД
13  ВСЕ;
14  КНЦ;

15 ФУНК DECBYN(BYN); (* 2 -> 10 *)
16 ИМЕНА: DEC,РАЗРЯД;
17 ЕСЛИ BYN ВИДА "0" ТО
18  0->DEC;

```

```

19  ДЛЯ РАЗРЯД ИЗ BYN ::
20  ЕСЛИ РАЗРЯД ИЗ "01" ТО
21  DEC+DEC->DEC;
22  ЕСЛИ РАЗРЯД="1" ТО DEC+1->DEC
23  ВСЕ
24  ИНАЧЕ
25  ?"ОШИБКА: DECBYN(",BYN,")"; ВЫХОД
26  ВСЕ
27  ИНАЧЕ
28  ?"ОШИБКА: DECBYN(",BYN,")"; ВЫХОД
29  ВСЕ;
30  РЕЗ:DEC;
31  КНЦ;

10  REM 10 -> 2
20  INPUT " ДЕСЯТИЧНОЕ: "; DEC:
30  A=DEC: BYN=""
40  B=INT(A/2)
50  IF A>B+B THEN BYN="1"+BYN;GOTO 60
60  BYN="0"+BYN; A=B
70  IF A>0 THEN 30
80  GOTO 20

10  REM 2 -> 10
20  INPUT " ДВОИЧНОЕ: "; BYN
30  DEC=0: L=LEN(BYN)
40  FOR I=1 TO L: R=MID(BYN,I,1)
50  IF NOT (R="1" OR R="0") THEN 20
60  DEC=DEC+DEC
70  IF R="1" THEN DEC=DEC+1
80  NEXT I
90  ?" ДЕСЯТИЧНОЕ: "; DEC
100 GOTO 20

```

К. В Рапире к функциям перевода можно обратиться из программы или в диалоге, например: `#:BYNDEC(1023);` или `?DECBYN(«10010»);`. Оператором ИМЕНА рабочие переменные A, B, BYN, DEC и РАЗРЯД локализованы в функциях, т. е. не доступны из других программ и не пересекаются с одноименными переменными в других программах. В строках 3 и 17 делается проверка на соответствие типа входного параметра — число или текст, а в строке 20 — на то, что текст состоит лишь из 0 и 1, при ошибке выдается соответствующее сообщение. Бейсик-программы зациклены, а в последнем случае при ошибке будет просто запрашиваться новое двоичное число, так как проверку типа на вводе Бейсик делает автоматически.

Р. Можно попробовать записывать двоичные числа не текстами, а десятичными, состоящими из 0 и 1. Запрограммируйте сложение и умножение двоичных чисел. Вспомнив тему «Коды и символы», исследуйте, как разные символы представлены в машине, как она различает русские и латинские буквы, большие и малые. Используя команды АДРЧ и АДРЗ (РЕЕК и РОКЕ), по адресам графической страницы разберитесь в их содержимом, попробуйте изменить цвет, сформировать изображение в цветном или черно-белом режиме.



## 21. Шестнадцатиричные числа

С. Двоичная запись все же труднообозрима, да и обрабатывает данные машина не побитно, а побайтно, т. е. группами по 8 бит. Поэтому освоен еще одну систему счисления — с основанием шестнадцать. В ней четыре двоичных разряда изображаются одним знаком — цифрой от 0 до 9 или латинской буквой А, В, С, D, Е, F, а любой байт — двумя знаками. Эта система по емкости превосходит десятичную, например: двузначной десятичной записи 15 соответствует шестнадцатиричная F, 255 — FF. В этой системе обычно адресуется память в Бейсике и Ассемблере.

```

1 ФУНК НЕХДЕС(ДЕС); (* 10 -> 16 *)
2 ИМЕНА:НЕХ,А,В;
3 ЕСЛИ ДЕС ВИДА 0 ТО
4   ""->НЕХ; ДЕС->А;
5   ПОКА А>0 :: А//16->В; А-16*В->А;
6   ЕСЛИ А>9 ТО А+7->А ВСЕ;
7   ВСЕ(А+176)+НЕХ->НЕХ; В->А
8   ВСЕ;
9 ИНАЧЕ ?"ОШИБКА: НЕХДЕС(",ДЕС,")";
                                     ВЫХОД
10 ВСЕ;
11 РЕЗ:НЕХ;
12 КНЦ;

```

```

13 ФУНК ДЕСНЕХ(НЕХ); (* 16 -> 10 *)
14 ИМЕНА:ДЕС,РАЗРЯД,Д;
15 ЕСЛИ НЕХ ВИДА "Ю" ТО
16   0->ДЕС;
17   ДЛЯ РАЗРЯД ИЗ НЕХ ::
18     ЕСЛИ РАЗРЯД ИЗ
19       "0123456789ABCDEF" ТО
20       КОД(РАЗРЯД)-176->Д;
21       ЕСЛИ Д>9 ТО Д-7->Д ВСЕ;
22       ДЕС*16+Д->ДЕС
23       ИНАЧЕ
24       ?"ОШИБКА: ДЕСНЕХ(",НЕХ,")";ВЫХОД
25       ВСЕ
26       ВСЕ
27 ИНАЧЕ
28 ?"ОШИБКА: ДЕСНЕХ(",ДЕС,")"; ВЫХОД
29 ВСЕ;
30 РЕЗ:ДЕС;
31 КНЦ;

```

```

10 REM 10 -> 16
20 INPUT "ДЕСЯТИЧНОЕ: ";ДЕС:
   А=ДЕС: НЕХ=" "
30 В=INT(А/16): А=А-В*16
40 IF А>9 THEN А=А+7
50 НЕХ=CHR$(А+176)+НЕХ: А=В
60 IF А>0 THEN 30
70 ? "ШЕСТНАДЦАТЕРИЧНОЕ: ";НЕХ
80 GOTO 20

```

```

10 REM 16 -> 10
20 INPUT "ШЕСТНАДЦАТЕРИЧНОЕ: ";НЕХ
30 ДЕС=0: L=LEN(НЕХ)
40 FOR I=1 TO L: R=MID$(НЕХ,I,1)
50 Д=ASC(R)-176

```

```

60 IF Д>1 THEN Д=Д-7
70 IF Д>15 OR Д<0 THEN 20
80 ДЕС=ДЕС*16+Д
90 NEXT I
100 ? "ДЕСЯТИЧНОЕ: ";ДЕС
110 GOTO 20

```

К. При обработке текстов с шестнадцатиричными числами пользуемся тем, что у символов от 0 до 9 коды от 176 подряд до 185, а от А до F — 193—198. Вообще говоря, в Бейсике перевод 16→10 не очень актуален, так как предусмотрена шестнадцатиричная форма представления констант, и для чисел меньше 8000 можно воспользоваться, например, оператором печати: ? 2С3А. Попробуйте, что получится при переходе через 8000.

Р. Запрограммируйте функции НЕХВУН (2→16) и ВУННЕХ (16→2). Сделайте программу перевода чисел из системы с основанием М в систему с основанием N, где М и N можно задать любыми, а в качестве цифр свыше 9 используются латинские буквы.

63

## 22. Сборка Бейсик-программ, или Как сделана эта статья

Занявшись составлением сложных программ на Бейсике, вы вскоре столкнетесь с необходимостью использовать в них фрагменты других программ, например, создание графиков, сложные вычисления или музыкальные коды. Хранение таких отлаженных подпрограмм в А-файлах ничего не даст, так как при считывании по команде LOAD прежняя программа пропадает. Набирать такие куски с распечатки заново — непроизводительная трата сил и времени с неизбежными ошибками и лишней отладкой. Выход в использовании Т-файлов. Обычно мы вводим тексты программ с клавиатуры, а вывод текстов делаем на экран или бумагу. Но можно вводить тексты программ с диска из Т-файла и «распечатывать» программу на диск — в Т-файл. Команды LOAD и SAVE не в счет, они работают с кодовой формой хранения Бейсик-программ в оперативной памяти и в А-файлах. Для того чтобы вывести текст программы в Т-файл, добавим в нее такую строку:

```

0 INPUT N$: ? : ?CHR$(4);"OPEN"+N$:
  ? : ?CHR$(4);"WRITE"+N$: LIST1,:
  ? : ?CHR$(4);"CLOSE"+N$: END

```

и, вставив диск, запустим измененную программу: RUN. На запрос введем имя для Т-файла. Записанный файл можно теперь прочитать в редакторе «Школьницы». Если хотите, Бейсик-программы можно сразу набирать в редакторе, с соответствующими

удобствами и неудобствами, и здесь же делать сборку, выбрав в меню ДОПИСАТЬ ФАЙЛ.

В Бейсике программа из Т-файла вводится по команде ЕХЕСИмя. Чтобы видеть читаемый текст, не забудьте перед этим ввести команду MON,1. Так, нетрудно собрать программу из нескольких Т-файлов, при этом следует иметь в виду, что если номера строк в них будут повторяться, то, как и при вводе с клавиатуры, последующая строка затрет в памяти предыдущую с таким же номером.

И вышеприведенную 0-строку тоже удобно хранить в отдельном Т-файле, например с именем W, вводя ее при необходимости по ЕХЕСW и не пользуясь номером 0 в программах для других целей.

А статья эта была набрана в системе «Агат-автор», которая умеет читать тексты Т-файлов, преобразовывать их в свой формат и записывать в виде В-файлов. Таким образом и были собраны и описаны все приведенные программы.

**С. ПАВЛОВ, Б. ЦИЛЕВИЧ**  
ВЦ при ЛатвГУ им. П. Стучки

## ППС на основе математического моделирования

Компьютеризация образования достаточно проста со снабженческой колокольни: нужно поставить (изготовить, купить, достать) в школы ЭВМ. Однако с функциональной точки зрения она подразделяется на два различных аспекта:

использование компьютера как объекта изучения;

использование компьютера как средства, способного повысить эффективность обучения определенному (предполагается — достаточно широкому) кругу предметов.

Сам по себе факт использования ЭВМ в учебном процессе, безусловно прогрессивный, отнюдь не обеспечивает повышения педагогической эффективности. Ключевой вопрос компьютеризации образования — методическая оправданность применения ЭВМ в изучении данного курса, темы, раздела.

Педагогическая наука и практика работали более или менее эффективные формы и методы обучения, адекватные составу и содержанию конкретного учебного курса. Естественно, что формы и методы обучения основаны на доступных — в период

## Литература

1. ПЭВМ «Агат»: Техническое описание. Ч. 2. Программное обеспечение. Фг3.032.002 ТО1. Москва, 1986.

2. Описание языка Рапира в системе «ШКОЛЬНИЦА». 3533847.0042-01 33 01. Новосибирск, 1987.

3. Степанов М. Е. Программирование на языке Бейсик-Агат: Методические рекомендации учителю Ч. I. НИИ школ МП РСФСР. Москва, 1986.

4. Самолысов В. А. Учебные задания по работе на ПЭВМ «Агат» (для учащихся девятых классов). М.: изд. НИИ школ МП РСФСР, 1986.

5. Уоттенберг Ф. Программирование для персонального компьютера Эпл: Пер. с англ. М.: Мир, 1988.

6. Алексеев М., Рябова А. «Агат», ассемблер и музыка // Информатика и образование. 1989. № 5.

на, в частности, и высокой производительностью труда разработчика подобных ППС: современный уровень развития системного программного обеспечения дает возможность работать на простых языках высокого уровня, реализованных даже на бытовых ЭВМ типа БК-0010. При этом основной упор переносится с принципиально важного этапа — разработки методики обучения с помощью ППС и одновременного проектирования сценария ППС — на чисто технический этап реализации обучающей программы.

Для обеспечения действительно эффективного использования ЭВМ в обучении необходимо силами группы специалистов (педагога, психолога, предметника, методиста, программиста, дизайнера и др.) разработать методику «машинного» обучения по данному конкретному курсу, исходя из целей обучения. При этом, весьма вероятно, придется пересмотреть состав и содержание предметного курса: во-первых, программа учебного курса связана с конкретной методикой; во-вторых, если, как ожидается, компьютеризация позволит повысить эффективность обучения, то изменится соотношение объема учебного материала и времени изучения предмета. На этом пути уменьшится «валовая» производительность труда разработчиков ППС, но повысится качество программного продукта.

### Математическое моделирование и обучение

Вопрос: где можно ожидать высокой педагогической эффективности ППС? — связан с другим: какие новые подходы и методики компьютеризация делает доступными для преподавателя?

Наибольшего внимания заслуживает метод математического моделирования (ММ), позволяющий проводить вычислительный эксперимент (ВЭ) [1] с помощью ЭВМ. Для изучения предметов естественнонаучного цикла ММ — наиболее адекватный подход. Вдобавок его последовательное и систематическое применение позволяет сблизить методологию учебной деятельности с методологией научно-исследовательской работы, дает обучаемому возможность освоить не только конкретный учебный курс, но и сам метод ММ, что особенно актуально для высшего образования и завершающей стадии среднего образования. Возможности применения ММ в обучении рассмотрены в ряде появившихся в последнее время работ [2, 3, 4].

ППС на базе метода ММ (назовем их ППСММ), видимо, должны иметь ряд

существенных отличий от пакетов прикладных программ (ППП), реализующих метод ММ. В отличие от исследователя, использующего ППП для получения объективно новых знаний, обучаемый с помощью ППСММ получает субъективно новые знания. Принципиально различается контингент конечных пользователей ППП и ППСММ. Традиционный пользователь ППП уже владеет «идеологией», а в большинстве случаев и «техникой» моделирования (т. е. участвует или может участвовать в разработке ППП). Для задач обучения овладение «идеологией» (в значительно меньшей степени — «техникой») моделирования может являться одной из функций ППСММ.

Перспективной предметной областью разработки и применения ППСММ являются предметные курсы, в которых метод ММ сам по себе широко применяется. Для завершающей стадии среднего образования многообещающими выглядят курсы физики и астрономии. В системе высшего образования перспективны разработка и применение учебно-исследовательских программных комплексов на базе ММ, которые могут применяться как в учебных, так и в научно-исследовательских целях. Их разработка и внедрение позволят не только повысить практическую отдачу студенческих работ, заинтересованность студентов, но и облегчить адаптацию молодых специалистов.

Внимательного рассмотрения требует вопрос о роли ППСММ на различных этапах системы непрерывного образования. Представляется, что применение ППСММ наиболее эффективно не на стадии формирования представлений о фундаментальных принципах отдельных дисциплин и характерных для них методологических подходах, а на этапе приложении этих принципов и подходов, т. е. роль ППСММ повышается по мере продвижения обучаемого по ступеням системы непрерывного образования. Так, в системе повышения квалификации специалистов такие программные средства могут быть основой всего учебного процесса.

Систематическое применение ППСММ должно привести к изменению соотношения между временем, отводимым на различные формы учебных занятий (возрастает удельный вес самостоятельной работы, практических и лабораторных занятий), а также к изменению традиционного характера этих форм.

Особое значение имеет вопрос о функциях преподавателя в условиях обучения, базирующегося на использовании ППСММ. По сравнению с существующими методи-

ками обучения изменение этих функций, видимо, будет существенным. Роль преподавателя будет приближаться к роли консультанта, научного руководителя учащихся, приобретающих в процессе учебного исследования субъективно новые знания.

### ППСММ

Схема взаимодействия функциональных элементов ППСММ показана на рисунке. Специфика учебного моделирования наиболее существенно проявляется не в принципиальной структуре метода ММ (желтый цвет), а в характере перехода от одного этапа к другому и источниках получения информации, необходимой для реализации того или иного этапа.

Рассмотрим стадию построения предметной модели. Построить предметную модель — значит выбрать те элементы реальной системы, которые будут включаться в рассмотрение, и те взаимосвязи между ними, которые будут учитываться. При традиционном применении математического моделирования в исследовательских целях этот этап, как правило, не автоматизирован и даже не формализован. Источниками исходной информации для построения предметной модели служат личный опыт исследователя и данные, приведенные в специальной литературе. Если мы хотим сделать этот этап доступным для пользователя ППСММ, то должны, во-первых, подготовить библиотеку описаний элементов изучаемой системы и библиотеку описаний возможных связей между этими элементами, во-вторых, дать пользователю возможность диалоговыми средствами работать с этими библиотеками (например, в задачах динамики указать материальные точки или твердые тела, входящие в рассматриваемую систему, качественно описать взаимодействие между ними — трение, упругие силы, силы гравитационного притяжения и т. п.), в-третьих, желательна иметь в составе программного комплекса инструментальные средства, позволяющие пополнять эти библиотеки.

Подобная ситуация характерна и для этапа построения математической модели; здесь речь идет о библиотеках уравнений параметров элементов, уравнений взаимосвязей (например, для описания электрической схемы — уравнения резистора, конденсатора, диода и т. п., уравнения Кирхгофа), а также уравнений дополнительных (начальных и граничных) условий.

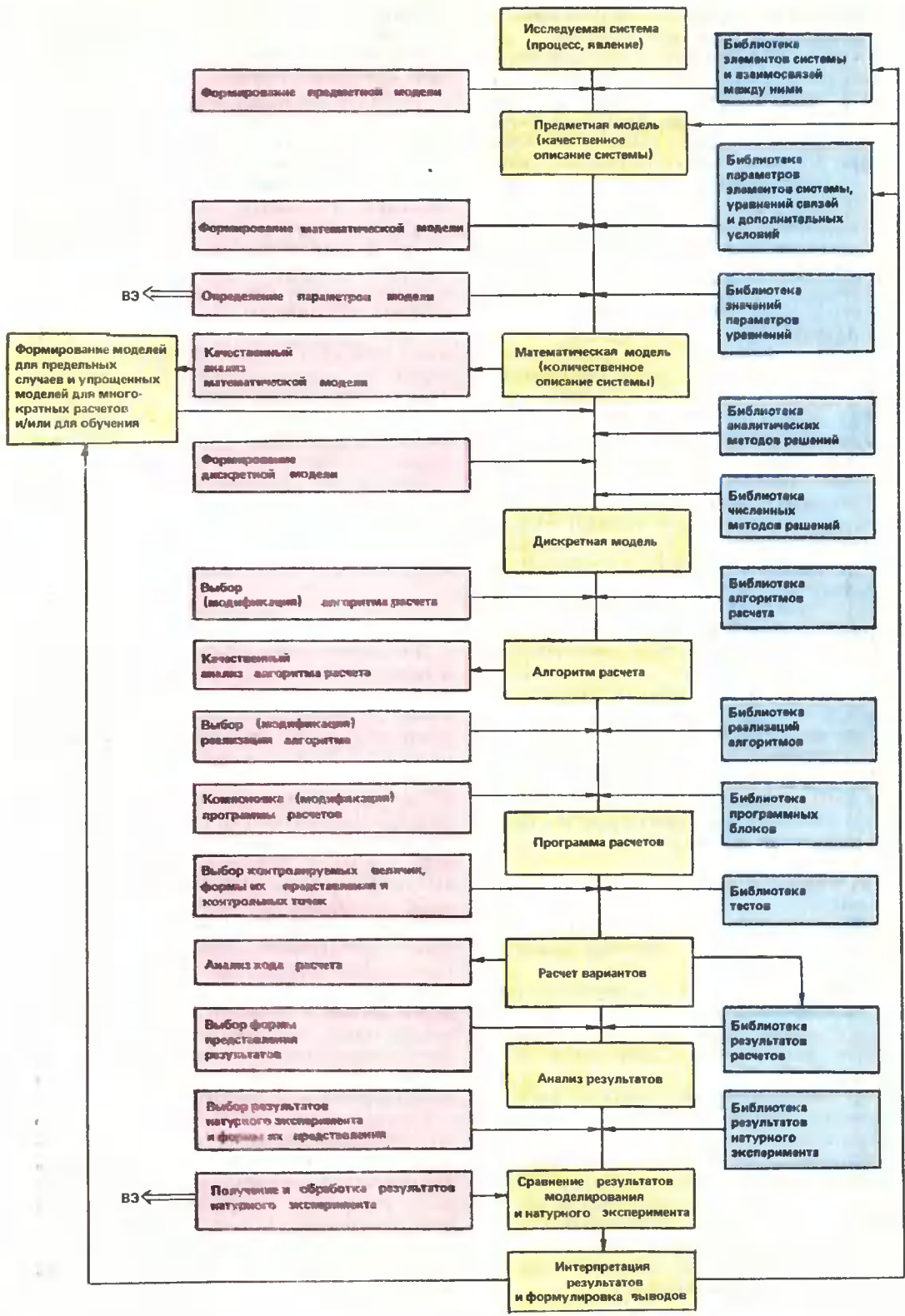
Весьма важен этап определения численных значений параметров математической модели. На практике эти параметры получают различными путями: прямым измерением,

косвенным измерением с последующими вспомогательными расчетами, иногда требуется вспомогательный вычислительный эксперимент, применяется даже подбор, или подгонка. Широкие возможности в этом отношении необходимо предоставить и пользователю ППСММ.

Наиболее существенно специфика учебного моделирования сказывается в способах получения, подачи обучаемому и использования результатов натурального эксперимента. Все данные, необходимые для интерпретации результатов доступных пользователю численных экспериментов, должны быть заранее подготовлены в соответствующих библиотеках, а также должны быть обеспечены возможности их представления в форме, наиболее удобной для интерпретации результатов моделирования.

Вообще, с точки зрения обучения, наиболее информативными этапами являются построение модели и интерпретация результатов. Использование на этих этапах средств визуализации (таблица, график, диаграмма, рисунок, динамическое изображение и т. п.), предоставление пользователю возможности выбора наиболее удобного из этих средств определяют степень реализации из дидактического принципа наглядности.

Функциональная конфигурация ППСММ в значительной мере определяется статусом пользователя, в зависимости от которого разрешается доступ к различным видам деятельности, обеспечиваемым программой (блоки, отмеченные на рисунке красным цветом). Так, пользователь со статусом разработчика имеет возможность модифицировать сам пакет, со статусом преподавателя или исследователя — пополнять библиотеки (блоки, отмеченные на рисунке синим цветом) коллективного пользования. Пользователя со статусом начинающего (например, младшего школьника, не владеющего «идеологией» моделирования) бессмысленно заставлять самостоятельно работать с моделью и алгоритмами, поэтому построение предметной и математической модели, выбор алгоритма, расчет происходят в этом случае автоматически, результаты представляются в форме, заранее определенной преподавателем, т. е. большая часть этапов для пользователя «закрыты». По мере повышения квалификации обучаемого его статус изменяется, он получает дополнительные возможности — например, изменять параметры алгоритма расчета, параметры модели, форму представления результатов и т. п. Такой подход позволяет «настраивать» программу на конкретного пользователя с учетом его реального уровня квалификации, при этом режим ППП является



ся предельным случаем, когда пользователю доступны все возможности ППСММ.

Со статусом пользователя связан и вопрос о выборе средства диалога. Наиболее простое средство — меню — оптимально для неопытного пользователя, имеющего низкий статус. Использование командного языка требует более высокой квалификации. Общение с программой на ограниченном естественном языке является наиболее перспективной формой диалога, обеспечивает наиболее широкие возможности. Кстати, необходимо предусмотреть возможность диалога обучаемого с ППСММ не только на русском, но и на других языках народов СССР. Для этого ППСММ должно содержать инструментальные средства модификации комплекса программ, обеспечивающие перевод в диалоговом режиме «текстовых констант», содержащихся в соответствующих программных блоках.

68

По-видимому, далеко не все функции ППСММ можно реализовать, используя традиционную организацию программного средства. Дидактический принцип индивидуализации обучения требует достаточно гибкой подстройки алгоритма к конкретному пользователю. Не исключено, что на определенных этапах работы обучаемого оптимальным вариантом сообщения ППСММ может быть: «Обратитесь к преподавателю (учебнику, справочнику) за консультацией». Но в основном организация диалога базируется на применении техники искусственного интеллекта. По существу, ППСММ представляет собой расчетно-логическую систему [5]. Выводы, полученные в результате вычислительного эксперимента при работе с программным комплексом, в формализованном виде могут пополнять базу знаний.

Структурно в составе ППСММ можно выделить:

моделирующее ядро (формирование моделей, аналитическое и численное решение, визуализация результата и др.);

программную оболочку, включающую элементы искусственного интеллекта (естественно-языковой интерфейс как средство диалога, база знаний о предметной области и др.);

инструментальные средства (модификация программ, пополнение библиотеки коллективного пользования и др.);

программы, обеспечивающие накопление, хранение, статистическую обработку и документирование информации о работе обучаемых (по типу автоматизированных обучающих систем нового поколения) [6];

программные средства поддержки локальной сети (при необходимости).

Кроме того, ППСММ должно включать методическое обеспечение (неважно, на каком носителе) в виде пособий в первую очередь для преподавателя.

Основное технологическое требование к ППСММ — модульность. Только она может обеспечить, с одной стороны, стандартизацию, необходимую для широкого и эффективного использования ППСММ в учебном процессе, а с другой стороны — возможность варьировать сценарий обучения. Соблюдение принципа модульного построения программ позволяет также использовать отдельные модули самостоятельно на менее мощных компьютерах.

Серьезные проблемы в обеспечении учебных заведений вычислительной техникой (вряд ли можно надеяться на решение этих проблем в ближайшее время) приводят к необходимости обеспечивать переносимость разрабатываемых ППС на другие виды техники. Поэтому для реализации программ целесообразно использовать структурные языки программирования (например, Паскаль, Си).

## Выводы

Последовательное применение метода ММ и вычислительного эксперимента при изучении естественнонаучных курсов является одним из наиболее перспективных направлений разработки дидактически эффективных ППС. Использование ППСММ может привести к существенному пересмотру состава и содержания соответствующих учебных курсов на всех ступенях непрерывного образования.

В процессе общения пользователя с ППСММ при формировании моделей, анализе и интерпретации результатов, реализации других этапов моделирования необходимо использование элементов техники искусственного интеллекта. Для обеспечения индивидуализации общения при обучении или исследовании в зависимости от статуса и квалификации пользователя целесообразно применение разнообразных форм организации диалога — «меню», командного языка, ограниченного естественного языка.

Построение моделирующего ядра ППСММ по модульному принципу из унифицированных программных блоков позволяет организовать их поэтапную разработку, автономное использование или последующее включение в интегрированный комплекс ППСММ с гибкой компоновкой блоков в соответствии с задачами учебного или исследовательского моделирования.

1. Самарский А. А. Математическое моделирование и вычислительный эксперимент // Вестник АН СССР. 1979. № 5.

2. Буланова Н. Л., Волков Д. В., Хозиев В. Б., Ширков П. Д. Математическое моделирование и перспективы развития школьного образования. М.: ИПМ АН СССР, 1987.

3. Фокин М. Л. Дидактические требования к учебным моделирующим программам на ЭВМ // Основные аспекты использования информационной технологии обучения в совершенствовании методической системы

обучения. М.: НИИ СиМО АПН СССР, 1987.

4. Павлов С. И., Цилевич Б. Л. Возможности применения метода математического моделирования для разработки педагогических программных средств // ЭВМ в образовании. Программное обеспечение. Рига: ЛатвГУ им. П. Стучки, 1988.

5. Поспелов Г. Искусственный интеллект — основа новой информационной технологии // Коммунист. 1988. № 1.

6. Довгялло А. М., Ющенко Е. Л. Обучающие системы нового поколения // УСиМ. 1988. № 1.

**Р. САЛИМЖАНОВ**

г. Петропавловск

## Стиль программирования

Как в ныне действующих школах учебных пособиях, так, наверное, и в будущих учебниках по ОИВТ элементам программирования будет уделяться достаточно большое внимание, так как программирование — это составная часть информатики.

В связи с этим полезно знакомить школьников с некоторыми приемами оформления программ. Это необходимо и с точки зрения развития общеобразовательных навыков, так как в каждой области знаний имеется набор приемов оформления результатов работы, которыми должны владеть специалисты.

Продуктом деятельности программиста является программа, которую, как правило, читают не только ее автор; поэтому люди, пишущие программы (в частности школьники), должны уметь читать и разбирать не только свои программы, но и программы своих товарищей.

Программы должны составлять так, чтобы их без значительных затруднений мог читать человек, а не машина, так как программы — это документ для дальнейшего использования, обучения и основа для создания более совершенных программ. Отсюда следует, какое важное значение имеют стиль написания программ, простота и доступность их текстов — даже за счет менее важных критериев, например краткости или машинной эффективности.

Под стилем программирования принято понимать набор приемов и методов программирования, которые применяют опытные специалисты для модификации, легко читаемых и доступных для понимания другими

программ. Поэтому правила хорошего стиля программирования — это результат соглашения между программистами; когда программист усвоит определенный стиль программирования, то его программы станут значительно легче для восприятия.

Учитывая, что учащиеся пишут, как правило, короткие программы, мы рассмотрим не все из существующих приемов написания программ в хорошем стиле, а только часть из них. С остальными правилами можно познакомиться по специальной литературе.

Рассмотренные в статье примеры написаны на языке Рапира. При желании учитель может использовать эти методические рекомендации в процессе преподавания основ программирования на базе таких языков, как Паскаль, Алгол, и даже таких «неструктурных» языков, как Бейсик, Фокал.

Отступы от начала строки

### Пример 1

```
Текст без отступов:
1->ФАК; ЕСЛИ И>1 ТО ДЛЯ К ОТ 1
      ДО И: ФАКЖК->ФАК ВСЕ ВСЕ;
Текст с отступами:
1->ФАК;
ЕСЛИ И>1
  ТО ДЛЯ К ОТ 1 ДО И:
    ФАКЖК->ФАК
  ВСЕ
ВСЕ;
```

Отступы, не оказывая влияния на логику программы, улучшают ее читаемость. Они аналогичны разделению обычного текста на абзацы, параграфы, главы.

### Пример 2

```

Текст без отступов:
ЕСЛИ A>B ТО A->MAX;
B->MIN ИНАЧЕ B->MAX;
A->MIN ВСЕ;
Текст с отступами:
ЕСЛИ A>B
  ТО A->MAX;
  B->MIN
  ИНАЧЕ B->MAX;
  A->MIN
ВСЕ;

```

Следует отметить, что правильно сделанные отступы не только улучшают восприятие текста программы, но и выявляют ее структуру, что полезно при дальнейшей работе с ней.

Отступы следует делать тогда, например, когда группа предписаний расположена между ключевыми словами ЕСЛИ, ТО, ИНАЧЕ.

### Пример 3

```

Текст без отступов:
ЕСЛИ A>B ТО ЕСЛИ A>C ТО
A->MAX ИНАЧЕ C->MAX ВСЕ
ИНАЧЕ ЕСЛИ B>C ТО B->MAX
ИНАЧЕ C->MAX ВСЕ ВСЕ;
Текст с отступами:
ЕСЛИ A>B
  ТО ЕСЛИ A>C
    ТО A->MAX
    ИНАЧЕ C->MAX
  ВСЕ
  ИНАЧЕ ЕСЛИ B>C
    ТО B->MAX
    ИНАЧЕ C->MAX
  ВСЕ
ВСЕ;

```

Группировки и отступы в примерах 2 и 3 помогают осмыслить сущность решаемых задач. Поэтому нетрудно понять, что в примере 2 переменные MAX и MIN получают наибольшее и наименьшее из значений величин A и B, а в примере 3 переменной MAX будет присвоено наибольшее из значений A, B и C.

Размещение нескольких длинных предписаний в одной строке нецелесообразно, так как за длинными выражениями, состоящими из большого числа нетривиальных частей, нелегко проследить. Кроме того, чем длиннее выражение, тем более вероятно появление в них ошибок. Так, например, запись программы в виде

```

ПРОЦ КОРНИ; 2->X; ПОКА X/=0:: ВВОД
  ДАННЫХ X; ЕСЛИ X>=0 ТО
?SQRT(X) ИНАЧЕ ?"ОШИБКА ВВОДА: ",X,
  "<0" ВСЕ ВСЕ КНЦ;

```

не отражает ее логику и затрудняет понимание. Поэтому предписания в программе должны иметь уровни, задаваемые величиной отступа от левого края записи каждого предписания, например так.

```

ПРОЦ КОРНИ;
2->X;
ПОКА X/=0::
  ВВОД ДАННЫХ X;
  ЕСЛИ X>=0
    ТО ?SQRT(X)
    ИНАЧЕ ?"ОШИБКА ВВОДА: ",X,"<0"
  ВСЕ
ВСЕ
КНЦ;

```

На этом примере можно еще раз убедиться в том, что такой подход помогает сразу увидеть предписания, которые входят в цикл (насколько труднее их увидеть по предыдущей записи!). Условное предписание здесь, как и предписание цикла, также имеет уровни, причем к внутреннему уровню относятся два альтернативных предписания (?SQRT(X) и ?«ОШИБКА ВВОДА: «,X,»<0»).

Рассмотрим наиболее часто встречающиеся примеры ошибок, которые допускают учащиеся при записи решений простейших задач.

Вернемся еще раз к циклам — типичному случаю, когда надо использовать отступы. При этом группировка в предписаниях цикла поможет определить начало и конец цикла, особенно во вложенных циклах.

### Пример 4

```

Текст без отступов:
0->C; 1->A; ПОКА A<=N::
C+A->C; A+1->A ВСЕ;
Текст с отступами:
0->C; 1->A;
ПОКА A<=N::
  C+A->C;
  A+1->A
ВСЕ;

```

### Пример 5

```

Текст без отступов:
0->C; 1->A; ПОВТОР 200::
C+A->C; A+1->A ВСЕ;
Текст с отступами:
0->C; 1->A;
ПОВТОР 200::
  C+A->C;
  A+1->A
ВСЕ;

```

### Пример 6

```

Текст без отступов:
""->НАОБОРОТ; ДЛЯ ЗНАК ИЗ
ТЕКСТ:: ЗНАК+НАОБОРОТ->
НАОБОРОТ ВСЕ;
Текст с отступами:
""->НАОБОРОТ;
ДЛЯ ЗНАК ИЗ ТЕКСТ::
  ЗНАК+НАОБОРОТ->НАОБОРОТ
ВСЕ;

```

### Пример 7

```

Текст без отступов:
0->ПОЛ; 0->ОТР; 0->НУЛИ;
ДЛЯ M ОТ 1 ДО N:: ДЛЯ
K ОТ 1 ДО N:: ВЫБОР ИЗ
КОР[M,K]<0:ОТР+1->ОТР;
КОР[M,K]>0:ПОЛ+1->ПОЛ
ИНАЧЕ НУЛИ+1->НУЛИ

```



```

ВСЕ ВСЕ ВСЕ;
Текст с отступами:
Ø->ПОЛ; Ø->ОТР; Ø->НУЛИ;
  ДЛЯ М ОТ 1 ДО N:
    ДЛЯ К ОТ 1 ДО N:
      ВЫБОР ИЗ
        КОР(М, К) < Ø: ОТР+1->ОТР;
        КОР(М, К) > Ø: ПОЛ+1->ПОЛ
        ИНАЧЕ НУЛИ+1->НУЛИ
      ВСЕ
    ВСЕ
  ВСЕ;

```

На практике обычно используют последовательные отступы на две-три позиции. Этого расстояния вполне достаточно, чтобы указать абзац и сделать несколько уровней отступа.

### Выбор имен переменных

Использование осмысленных имен переменных является одним из необходимых приемов программирования. Очень трудно, например, разобраться с программой, в которой есть имена вида ХХ, ХХХ, ХХХХ, ХУ, УХ, ЕКК, ЕККК. Эти имена ничего не говорят о своем предназначении. Чтобы как-то поправить дело, программы с такими именами заполняют комментариями, разъясняющими смысл и назначение этих имен. Использование же более длинных, но содержательных имен может существенно облегчить чтение программы и свести к минимуму количество необходимых комментариев. Однако слишком длинные имена, типа ЧИСЛО\_ТОЧЕК\_В\_ГРАФЕ\_РАССТОЯНИЙ, также нежелательны, потому что они не только отвлекают внимание от программы, но и таят возможность ошибок. Например, в предписании присваивания  $Y * X \rightarrow Z$  имена переменных выбраны неудачно, поскольку совсем не использована мнемоника. Такая запись, как  $I * R \rightarrow U$  или СИЛА\_ТОКА \* СОПРОТИВЛЕНИЕ  $\rightarrow$  НАПРЯЖЕНИЕ несет больше информации.

Следует понимать, что правильный выбор имен переменных — это залог удобочитаемости программ.

Необходимо учесть, что от имен, схожих по виду, составу или звучанию (ВАРИАНТЫ, ВАРИАЦИЯ, ВАРИАНТ, ВАРИАЦИИ, ТЕКСТ12, ТЕКСТ21, ТЕКСТ, ТЕКТ и т. д.), мало пользы. Поэтому выбирать осмысленные имена нужно так, чтобы их было трудно перепутать.

Если в именах используются цифры, то лучше помещать их в конце. Кроме того, надо помнить, что цифры 0, 1, 7 и 5 легко спутать с буквами O, I, Z и S.

```

Пример 8
(Ж СОКРАЩЕНИЕ ОТНОШЕНИЯ М К Н Ж)
ПРОЦ СОКР (<=>М, <=>Н);
ИМЕНА: М, Р;
М->П; Н->Р;

```

```

ПОКА П/=Р:;
  ЕСЛИ П>Р ТО П-Р->П ВСЕ
  ЕСЛИ П<Р ТО Р-П->Р ВСЕ
ВСЕ
М/П->М; Н/П->Н
КНЦ;

```

#### Пример 9

```

ФУНК ЗНАК;
ИМЕНА: Р;
"+"->Р;
ЕСЛИ ДСЧ() < Ø.5
  ТО "-"->Р
ВСЕ;
РЕЗ: Р
КНЦ;

```

#### Пример 10

```

ФУНК СЛ_ЧИСЛО(Н, К);
РЕЗ: ЦЕЛЧ(ДСЧ()) * (К+1-Н)
КНЦ;

```

Соответствующая мнемоника также важна и при выборе имен для процедур, функций и файлов. В качестве имен желательно употреблять термины, используемые в той отрасли, для которой пишется программа. Например, в процедурах и функциях примеров 8—10 имена СОКР, ЗНАК и СЛ\_ЧИСЛО напоминают учащимся о предназначении этих функций и процедур.

### Комментарии

Полезность комментариев в тексте программы, казалось бы, очевидна. Однако далеко не все учащиеся используют их. Они считают, что комментарии можно опускать с целью экономии времени при составлении программы. Однако даже через достаточно короткое время и сами авторы программы забывают многие ее детали. В этом случае вся первоначальная «экономия» не сравнима с дальнейшими затратами времени, превышающими ее во много раз.

Комментарии лучше составлять в процессе написания программы, так как именно в это время учащиеся в наибольшей степени вникают во все детали. Цель комментариев — облегчить понимание программы, следовательно, они должны быть так же хорошо продуманы, как и план решения задачи.

Так, в примере 8 комментарий (\* СОКРАЩЕНИЕ ОТНОШЕНИЯ М К Н \*) содержит полезную, раскрывающую цель соответствующей процедуры информацию. Рассмотрим ряд других небольших программ и фрагментов программ с комментариями.

Обратите внимание на то, что все комментарии в примере 11 смещены вправо и начинаются в одной колонке. Этот прием улучшает наглядность программы, так как учащиеся могут просматривать взаимосвязанные части программы, не прерываемые комментариями. В этом случае коммента-

**Пример 11**  
 ПРОЦ СУММА (S=>);  
 0->S; (Ж НАЧАЛЬНЫЕ ЗНАЧЕНИЯ СУММЫ S Ж)  
 1->N; (Ж И ТЕКУЩЕГО СЛАГАЕМОГО N Ж)  
 ПОКА NK=100; (Ж НАРАЩИВАНИЕ СУММЫ Ж)  
 S+N->S; (Ж ВЫЧИСЛЕНИЕ СЛЕДУЮЩЕГО СЛАГАЕМОГО Ж)  
 N+1->N;  
 ВСЕ  
 КНЦ;

рии в программе аналогичны подстрочным примечаниям в книге.

Следует обязательно учитывать, что комментарии должны содержать некоторую информацию о логике построения программы, а не перефразировать то, что и так ясно из текста. Не следует писать такие комментарии, как в примере 12. Они лишь засоряют текст, объясняя синтаксис языка программирования, а не логику программы. Очевидно, что для примера 12 более подходящим будет, например, комментарий вида (• УВЕЛИЧИТЬ СЧЕТЧИК КРАСНЫХ ТОЧЕК \*).

72

**Пример 12**  
 СЧЕТ+1->СЧЕТ (ЖУВЕЛИЧИТЬ СЧЕТ НА 1 Ж)

Однако следует помнить, что в отдельном предписании довольно редко проявляются какие-либо логически законченные действия, поэтому нет необходимости комментировать каждое предписание. Удобно использовать комментарии для разъяснения смысла относительно крупных разделов программы.

**Пример 13**  
 (Ж-----Ж)  
 (Ж ЗАДАТЬ КОЛИЧЕСТВО ВАРИАНТОВ Ж)  
 (Ж-----Ж)  
 ПОЗ(0,3);  
 ?"СКОЛЬКО ХОТИТЕ ВАРИАНТОВ?";  
 ВВОД ДАННЫХ: VAR;  
 (Ж-----Ж)  
 (Ж ОПРЕДЕЛЕНИЕ ШИФРА ИЗ НЕ БОЛЕЕ Ж)  
 (Ж ЧЕМ СЕМИ СИМВОЛОВ Ж)  
 (Ж-----Ж)  
 "2222111111"->ШИФР;  
 ПОКА #ШИФР>7;:  
 ПОЗ(0,3);  
 ?" ВВЕДИТЕ ШИФР РАБОТЫ";  
 ?" (НЕ БОЛЕЕ СЕМИ СИМВОЛОВ)";  
 ПОЗ(0,3);?" Ж"; ПОЗ(0,3);  
 ВВОД: ШИФР;  
 ВСЕ;

Заключение комментариев в прямоугольник из специальных символов позволяет отделять их от основного текста программы и не мешает ее чтению.

При написании достаточно больших программ нужны соответственно и более подроб-

ные комментарии, отражающие, в частности, название или краткое описание используемого алгоритма, входные и выходные параметры, назначение основных переменных и вызываемых подпрограмм.

Рассмотрим один из таких примеров.

**Пример 14**  
 ПРОЦ ПЛОЩАДЬ (A, B, N, MAX, S=>);  
 (Ж-----Ж)  
 (Ж ВЫЧИСЛЕНИЕ ПЛОЩАДИ КРИВОЛИНЕЙНОЙ Ж)  
 (Ж ТРАПЕЦИИ МЕТОДОМ МОНТЕ-КАРЛО Ж)  
 (Ж ПАРАМЕТРЫ: Ж)  
 (Ж A, B - КОНЦЫ ОСНОВАНИЯ ТРАПЕЦИИ; Ж)  
 (Ж N - ЧИСЛО "БРОСАЕМЫХ" ТОЧЕК; Ж)  
 (Ж MAX - НАИБОЛЬШЕЕ ЗНАЧЕНИЕ Ж)  
 (Ж ДАННОЙ ФУНКЦИИ; Ж)  
 (Ж S - РЕЗУЛЬТАТ. Ж)  
 (Ж ОСНОВНЫЕ ПЕРЕМЕННЫЕ: Ж)  
 (Ж СЧЕТ - ЧИСЛО ТОЧЕК, ПОПАВШИХ В Ж)  
 (Ж ТРАПЕЦИЮ. Ж)  
 (Ж ВЫЗЫВАЕМЫЕ ФУНКЦИИ: Ж)  
 (Ж СЛ\_ЧИСЛО(N,K) - ВОЗВРАЩАЕТ СЛУ- Ж)  
 (Ж ЧАЙНОЕ ЧИСЛО ОТ N ДО K; Ж)  
 (Ж F(X) - ВЫЧИСЛЯЕТ ЗНАЧЕНИЕ Ж)  
 (Ж ФУНКЦИИ. Ж)  
 (Ж-----Ж)  
 ИМЕНА: K, СЧЕТ, X, Y;  
 0->СЧЕТ;  
 (Ж-----Ж)  
 (Ж ВЫЧИСЛЕНИЕ КОЛИЧЕСТВА ТОЧЕК, Ж)  
 (Ж ПОПАВШИХ В ТРАПЕЦИЮ Ж)  
 (Ж-----Ж)  
 ДЛЯ K ОТ 1 ДО N;:  
 СЛ\_ЧИСЛО(A, B)->X;  
 СЛ\_ЧИСЛО(0, MAX)->Y;  
 ЕСЛИ Y<=F(X)  
 ТО СЧЕТ+1->СЧЕТ  
 ВСЕ  
 ВСЕ;  
 (Ж-----Ж)  
 (Ж НАХОЖДЕНИЕ ПЛОЩАДИ ПО МЕТОДУ Ж)  
 (Ж МОНТЕ-КАРЛО Ж)  
 (Ж-----Ж)  
 СЧЕТ/NЖ(B-A)ЖMAX->S  
 КНЦ;

ФУНК СЛ\_ЧИСЛО(N,K);  
 РЕЗ: ЦЕЛЧ(ДСЧ())Ж(K-N+1)))+N  
 КНЦ;

ФУНК F(X);  
 РЕЗ: XЖX2  
 КНЦ;

## Графика на алфавитно-цифровом дисплее: реализация на Фортране

При формировании графической информации на экране алфавитно-цифрового дисплея (АЦД) широко применяется метод построения с помощью единичных графических элементов (ЕГЭ), положение которых задается в операторах вывода. Так, в [1] рассмотрен подход к обучению элементам программирования на основе решения графических задач на Бейсике и Фортране. Однако положения ЕГЭ в указанной работе задаются без учета функциональных возможностей дисплея, его режимов, что требует разработки отдельных подпрограмм управления маркером, которые характеризуются сложностью, громоздкостью, необходимостью знания макроассемблера. В работе [2] предложена более простая методика организации вывода графической информации на экране АЦД с применением системы команд № 2, однако разработанные программы ориентированы только на Бейсик и не могут быть перенесены на Фортран без знания особенностей как самого языка, так и режимов работы АЦД. Данная статья заполняет пробел и позволяет предложенную в [2] методику распространить на Фортран в операционной системе СМ РАФОС.

В общем виде для Фортрана оператор вывода информации на экран АЦД в системе команд № 2 (см. [2]) записывается так:

```
WRITE(5,1) I1, I2, ..., IN
1 FORMAT(X, NA1)
```

где 5 — номер устройства вывода (АЦД); 1 — форматная метка; I1, I2, ..., IN — список кодов, в качестве которых используются десятичные коды командных последовательностей символов, коды положений маркера или коды символов КОИ-7; X — спецификация для управления маркером на экране АЦД; A — спецификация для вывода символов в коде КОИ-7; N — повторитель. Таким образом, для реализации команды «Прямая адресация маркера» на Фортране не требуются громоздкие подпрограммы управления маркером, как в [1]. Достаточно использовать, например, следующий набор операторов:

```
WRITE(5,1) 27, 89, 31+Y, 31+X
1 FORMAT(X, 4A1)
```

Он переместит маркер по экрану в точку с координатами (X, Y), где Y=1,24 — номер строки; X=1,80 — номер позиции в строке

для дисплея типа 15ИЭ-00-013. При этом предварительно необходимо произвести подготовку экрана АЦД к работе, выполнив команды «Переход к системе команд № 2», «Возврат» и «Стирание страницы» (см. [2]):

```
WRITE(5,1) 23, 27, 72, 27, 74
1 FORMAT(X, 5A1)
```

Приведенные в [2] методы формирования горизонтальных, вертикальных, наклонных линий (отрезков) и различных фигур сравнительно легко могут быть переведены на Фортран с учетом отмеченных особенностей. Примером может служить подпрограмма LINE (строки 2—8 приложения 3), позволяющая формировать на экране горизонтальные и вертикальные линии. Формальными параметрами подпрограммы являются X1, Y1, X2, Y2 — координаты начальной и конечной точек соответственно; EGE — код символа (ЕГЭ), формирующего изображение линии на экране АЦД. Организация горизонтальной линии осуществляется заданием параметра Y1, равным Y2, а вертикальной линии — заданием X1, равным X2. Обращение к подпрограмме производится после задания фактических параметров с помощью оператора

```
CALL LINE(X1, Y1, X2, Y2, EGE)
```

Аналогично программам, реализованным на Бейсике [2], можно строить различные статические изображения на экране АЦД с использованием Фортрана. В приложениях 1, 2 в качестве примера приведены программы формирования на экране АЦД спирали, эллипса.

В ряде случаев, например при разработке программ учебного или игрового характера (для автоматизированных учебных курсов, обучающе-контролирующих, игровых программ), возникает необходимость организации динамических изображений. Покажем особенности формирования динамических объектов на экране АЦД.

Для приближенной оценки значения определенного интеграла широко применяется метод прямоугольников, заключающийся в разбиении интервала интегрирования [A, B] на N частей и оценке значения интеграла как суммы площадей вписанных прямоугольников:

$$\int_A^B f(x) dx \approx \frac{B-A}{N} (S_0 + S_1 + \dots + S_{N-1}).$$

Начальный этап разработки обучающей программы, реализующей метод прямоугольников, заключается в разбиении экрана АЦД на ряд полей вывода — отдельных областей экрана, в которых формируется различная информация (рис. 1). В качестве границ полей можно использовать горизонтальные и вертикальные линии, построение которых

рассмотрено выше. Для данной задачи необходимо создать:

поле вывода 1 — для организации вывода координатных осей, подынтегральной функции (статические объекты) и моделирования механизма построения прямоугольников на экране (динамические объекты);

поле вывода 2 — для вывода пояснительного и обучающего текстов, ввода исходных значений параметров, вывода результатов (текущего значения шага, вычисляемой площади, окончательного результата оценки);

поле вывода 3 — для вывода пояснительного текста (информация о действиях для продолжения, завершения работы программы и т. д.).

Организация вывода необходимой информации в заданное поле вывода осуществляется выполнением команды «Прямая адресация маркера» (т. е. заданием начальной координаты поля вывода), а затем выводом требуемого текста (сопроводительного, пояснительного и т. д.). Например, операторы вывода

```
WRITE (5, 2) 27, 89, 31+2, 31+9
2 FORMAT (X, 4A1, 'МЕТОД ПРЯМОУГОЛЬНИКОВ')
```

произведут вывод информации, заключенной в апострофах оператора формата, в поле вывода, начинающееся с координат, задаваемых в команде «Прямая адресация маркера», т. е. с 9-й позиции 2-й строки.

В ряде случаев необходимо ввод значений задаваемых параметров осуществлять в отдельных отведенных для этого полях вывода без стирания построенных статических изображений. Для этого необходима фиксация (позиционирование) маркера в соответствующей позиции поля. Однако достаточно простой оператор ввода под управлением списка (неформатный ввод) автоматически возвращает маркер из отведенного поля в начало следующей строки, что приводит к наложению вводимой информации на изображение. Приведенный ниже набор операторов позволяет обойти указанное ограничение.

```
WRITE (5, 1) 27, 89, 31+9, 31+45
1 FORMAT ('N', 4A1)
READ (5, X) A
```

Он позволяет вводимое значение параметра А отобразить в поле вывода начиная с 45-й позиции 9-й строки путем фиксации маркера в точке с координатами (45, 9).

В основном поле (поле вывода 1) необходимо сформировать как статические объекты (координатные оси, сопроводительный текст, график подынтегральной функ-

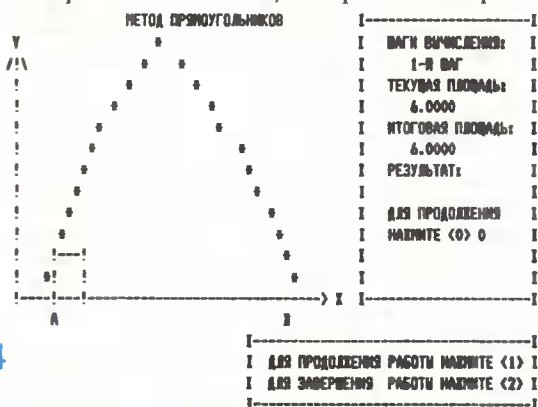


РИС. 1.А

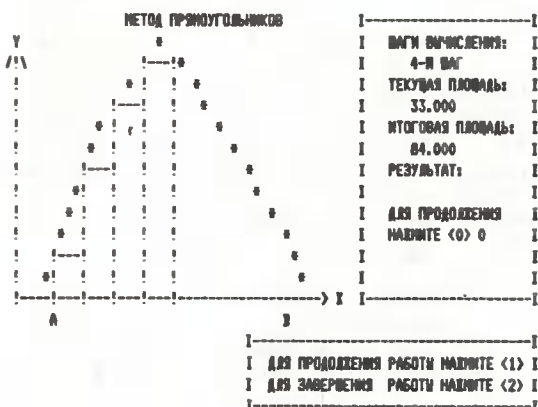


РИС. 1.Б

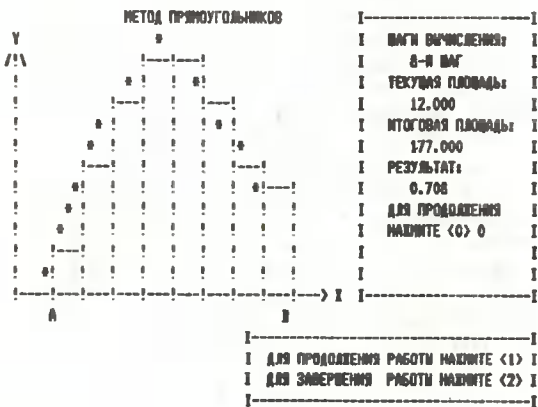


РИС. 1.С

ции), так и динамические объекты («мультипликация» механизма построения вписанных прямоугольников). Для этого следует на заданном интервале определить максимальное и минимальное значения функции, произвести масштабирование по горизонтали и вертикали, определить центр рисунка и произвести позиционирование его на экране АЦД, что позволит наиболее рационально распределить в поле вывода 1 отображаемый график функции. В более простом варианте допустимо отказаться от подобных операций и для конкретного вида функции, координаты которой можно задать в виде значений элементов массива, проиллюстрировать механизм оценки определенного интеграла на экране АЦД.

Построение координатных осей с сопроводительным текстом достаточно подробно изложено в [2], поэтому, используя подпрограмму LINE, можно организовать на экране АЦД горизонтальную ось с помощью ЕГЭ «—» и вертикальную ось с помощью ЕГЭ «!» (коды 45 и 33 соответственно).

Задание координат точек графика и формирование кривой подынтегральной функции в координатных осях поля вывода 1 производится операторами, идущими в приложении 3 вслед за комментарием «Формирование графика в поле вывода 1» (до оператора с меткой 18 включительно). Для формирования используется массив из 21 точки, которые можно отображать на экране АЦД с помощью любых ЕГЭ — например, звездочки (код 42) или точки (код 46).

Следующий этап — формирование прямоугольников в поле вывода 1. Этот процесс реализуется с помощью подпрограммы LINE, однако предварительно необходима начальная установка вычисляемой оценки.

Организация прямоугольников на экране АЦД производится с помощью трех операторов, следующих в приложении 3 вслед за комментарием, стоящим после метки 20.

Параллельно с построением прямоугольников целесообразно осуществить вычисление оценки определенного интеграла:

$$ST = (15 - Y(K(I))) * \Delta S$$

$$S = S + ST$$

Здесь ST — текущее значение площади прямоугольника;  $15 - Y(K(I))$  — высота прямоугольника; S — суммарное значение площади. При этом номер шага, текущее и суммарное (итоговое) значения вычисленных площадей выводятся в поле вывода 2 с учетом отмеченных особенностей Фортрана. Пошаговое восприятие учебного материала обеспечивается соответствующей задержкой путем нажатия, например, клавиши «0».

Таким образом, на примере разработки демонстрационно-обучающей программы, приведенной в приложении 3, рассмотрены особенности формирования графической информации на экране АЦД с применением языка Фортран.

## Л и т е р а т у р а

1. Буланов Н., Волков Д., Ширков П. Графика на алфавитном дисплее // Информатика и образование. 1987. № 3.
2. Карпов В., Карпова О., Новичков В. Графическая информация на алфавитно-цифровом дисплее // Информатика и образование. 1988. № 2.

ПРИЛОЖЕНИЕ 1  
ПОСТРОЕНИЕ СПИРАЛИ НА ЭКРАНЕ  
С ФОРМИРОВАНИЕМ СПИРАЛИ

```

INTEGER X,Y
WRITE (5,X) 'КООРДИНАТЫ X0,Y0'
READ (5,X) X0,Y0
DO 1 I=1,700
A=(I-1)*0.03
X=A*XCOS(A)+X0
Y=0.5*X*XSIN(A)+Y0
1 WRITE (5,2) 27,89,31+Y,31+X,42
2 FORMAT (X,5A1)
STOP
END

```

ПРИЛОЖЕНИЕ 2  
ПОСТРОЕНИЕ ЭЛЛИПСА НА ЭКРАНЕ  
С ФОРМИРОВАНИЕМ ЭЛЛИПСА

```

INTEGER X,Y
WRITE (5,X) 'ПОЛУОСИ A, B'
READ (5,X) A,B
WRITE (5,1) 'КООРДИНАТЫ X0, Y0'
READ (X,X) X0,Y0
WRITE (5,X) 'ЧИСЛО ТОЧЕК'
READ (5,X) N
DO 1 I=0,2*N
X=X0+A*XCOS(3.14*I/N)
Y=Y0+B*XSIN(3.14*I/N)
1 WRITE (5,2) 27,89,31+Y,31+X,42
2 FORMAT (X,5A1)
STOP
END

```

ПРИЛОЖЕНИЕ 3  
ДЕМОНСТРАЦИОННО-ОБУЧАЮЩАЯ ПРОГРАММА  
ВЫЧИСЛЕНИЯ ОПРЕДЕЛЕННОГО ИНТЕГРАЛА  
МЕТОДОМ ПРЯМОУГОЛЬНИКОВ

```

C ПОДПРОГРАММА ПОСТРОЕНИЯ ЛИНИИ
SUBROUTINE LINE (X1,Y1,X2,Y2,EGE)
INTEGER X1,Y1,X2,Y2,EGE
DO 1 J=X1,X2
DO 1 I=Y1,Y2
1 WRITE (5,2) 27,89,31+I,31+J,EGE
2 FORMAT (X,5A1)
RETURN
END
C ПОДГОТОВКА ЭКРАНА АЦД
27 WRITE (5,1) 23,27,72,27,74
1 FORMAT (X,5A1)
C ОРГАНИЗАЦИЯ ПОЛЯ ВЫВОДА 1
WRITE (5,2) 27,89,31+1,31+13
2 FORMAT (X,4A1,'МЕТОД ПРЯМОУГОЛЬНИКОВ')
C ФОРМИРОВАНИЕ ГРАНИЦЫ ПОЛЯ ВЫВОДА 2

```

```

CALL LINE (50, 1, 50, 15, 73)
CALL LINE (51, 1, 79, 1, 45)
CALL LINE (80, 1, 80, 15, 73)
CALL LINE (51, 15, 79, 15, 45)
C ФОРМИРОВАНИЕ ПОЯСНИТЕЛЬНОГО ТЕКСТА
WRITE (5, 4) 27, 89, 31+3, 31+54
4 FORMAT (X, 4A1, 'ШАГИ ВЫЧИСЛЕНИЯ: ')
WRITE (5, 5) 27, 89, 31+4, 31+57
5 FORMAT (X, 4A1, ' - Й ШАГ')
WRITE (5, 6) 27, 89, 31+5, 31+54
6 FORMAT (X, 4A1, 'ТЕКУЩАЯ ПЛОЩАДЬ: ')
WRITE (5, 7) 27, 89, 31+7, 31+54
7 FORMAT (X, 4A1, 'ИТОГОВАЯ ПЛОЩАДЬ')
WRITE (5, 8) 27, 89, 31+9, 31+54
8 FORMAT (X, 4A1, 'РЕЗУЛЬТАТ: ')
WRITE (5, 9) 27, 89, 31+11, 31+54
9 FORMAT (X, 4A1, 'ДЛЯ ПРОДОЛЖЕНИЯ')
WRITE (5, 10) 27, 89, 31+12, 31+54
10 FORMAT (X, 4A1, 'НАЖМИТЕ <0>')
C ОРГАНИЗАЦИЯ ПОЛЯ ВЫВОДА 3
C ФОРМИРОВАНИЕ ГРАНИЦЫ ПОЛЯ ВЫВОДА 3
CALL LINE (35, 19, 35, 22, 73)
CALL LINE (36, 19, 79, 19, 45)
CALL LINE (36, 22, 79, 22, 45)
CALL LINE (80, 19, 80, 22, 73)
C ФОРМИРОВАНИЕ ПОЯСНИТЕЛЬНОГО ТЕКСТА
WRITE (5, 11) 27, 89, 31+20, 31+37
11 FORMAT (X, 4A1, ' ДЛ Я ПРОДОЛЖЕНИЯ
Х РАБОТЫ НАЖМИТЕ <1>')
WRITE (5, 12) 27, 89, 31+21, 31+37
12 FORMAT (X, 4A1, ' ДЛ Я ЗАВЕРШЕНИЯ
Х РАБОТЫ НАЖМИТЕ <2>')
C ФОРМИРОВАНИЕ ОСЕЙ
CALL LINE (3, 15, 43, 15, 45)
CALL LINE (3, 4, 3, 15, 33)
WRITE (5, 13) 27, 89, 31+2, 31+354
13 FORMAT (X, 4A1, 'Y')
WRITE (5, 14) 27, 89, 31+4, 31+254
14 FORMAT (X, 4A1, '/!\')
WRITE (5, 15) 27, 89, 31+15, 31+44
15 FORMAT (X, 4A1, '> X')
WRITE (5, 16) 27, 89, 31+16, 31+8
16 FORMAT (X, 4A1, 'A', 31X, 'B')
C ФОРМИРОВАНИЕ ГРАФИКА В ПОЛЕ 1
INTEGER X(21), Y(21)
DATA X/8, 9, 10, 11, 12, 13, 14, 16, 18, 20,
X22, 24, 27, 28, 30, 33, 34, 35, 36, 38, 40/
DATA Y/13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3,
X4, 5, 6, 7, 8, 9, 10, 11, 12, 13/
DO 17 I=1, 21
17 WRITE (5, 18) 27, 89, 31+Y(I), 31+X(I), 42
18 FORMAT (X, 5A1)
C ФОРМИРОВАНИЕ ПРЯМОУГОЛЬНИКОВ
INTEGER K(8)
DATA K/1, 5, 8, 10, 12, 14, 16, 19/
S=0
DO 19 I=1, 8
C ВЫВОД В ПОЛЕ ВЫВОДА 2
WRITE (5, 20) 27, 89, 31+4, 31+56, I
20 FORMAT (X, 4A1, I2)
X8+(I-1)X4, 15, 33)
CALL LINE (8+(I-1)X4, Y(K(I)),
X8+(I-1)X4+4, Y(K(I)), 45)
CALL LINE (8+(I-1)X4+4, Y(K(I)),
X8+(I-1)X4+4, 15, 33)
C ВЫЧИСЛЕНИЕ ТЕКУЩЕЙ ПЛОЩАДИ
ST=(15-Y(K(I)))X3
C ВЫВОД В ПОЛЕ ВЫВОДА 2
WRITE (5, 21) 27, 89, 31+6, 31+60, ST
21 FORMAT (X, 4A1, 6I2.5)
C ВЫЧИСЛЕНИЕ ИТОГОВОЙ ПЛОЩАДИ
S=S+ST
C ВЫВОД В ПОЛЕ ВЫВОДА 2
WRITE (5, 22) 27, 89, 31+8, 31+60, S
22 FORMAT (X, 4A1, 6I2.5)
C ЗАДАНИЕ СЛЕДУЮЩЕГО ШАГА
WRITE (5, 23) 27, 89, 31+12, 31+66
23 FORMAT ('N', 4A1)
READ (5, X)
IF (C.NE.0) STOP
19 CONTINUE
C ВЫВОД В ПОЛЕ ВЫВОДА 2
WRITE (5, 25) 27, 89, 31+10, 31+60, S
25 FORMAT (X, 4A1, 6I2.5)
C ВЫВОД В ПОЛЕ ВЫВОДА 3
WRITE (5, 26) 27, 89, 31+21, 31+77
26 FORMAT ('N', 4A1)
READ (5, X) D
IF (D.EQ.1) GO TO 27
STOP
END

```

## П. ГУЖА

# Расширение возможностей КУВТ-86

Эксплуатация КУВТ-86 показала его достаточную эксплуатационную надежность. Созданное для него в НИИФТТ ЛГУ им. П. Стучки программное обеспечение (система «Рига») позволяет использовать класс и новичку для освоения вычислительной техники и других учебных предметов, и программисту-разработчику учебных программ.

Однако в ходе эксплуатации выяснились некоторые технические недостатки как отдельных устройств, так и всей системы в целом. Вот основные их них:

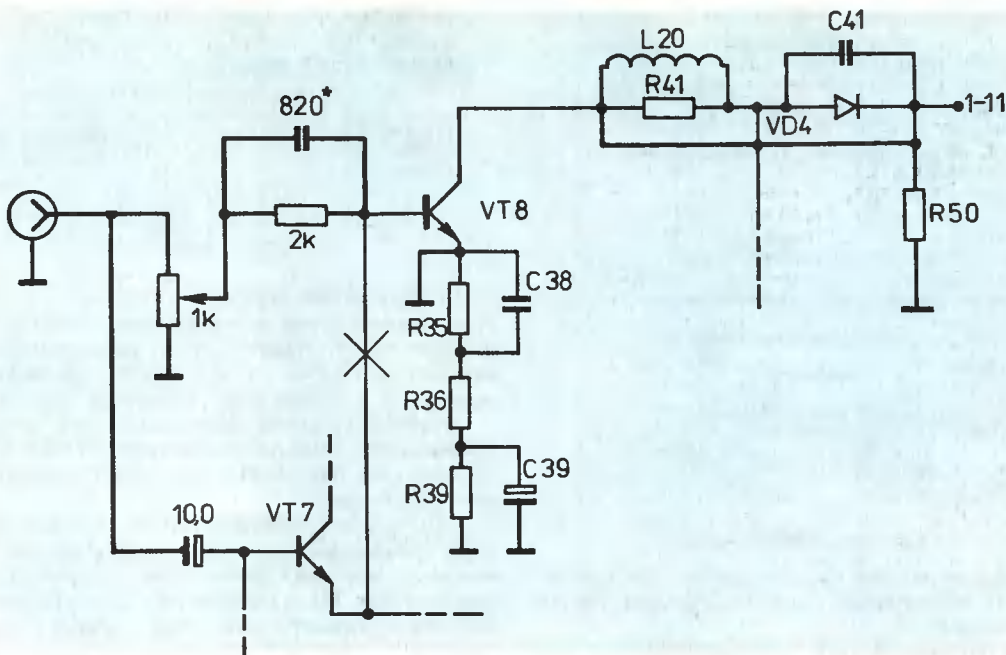
невысокое качество изображения на мониторе БК-0010;

ЭВМ на рабочих местах конструктивно не объединены в один узел, что снижает электробезопасность;

в случае выхода из строя ДВК-2М нет возможности передать информацию с другой ЭВМ или магнитофона на рабочие места;

Автором разработан ряд аппаратных средств, позволяющих устранить перечисленные недостатки и получить качественно новые возможности как для всего класса КУВТ-86, так и для отдельных БК-0010.

Улучшение монитора БК-0010. В качестве монитора БК-0010 используется бытовой телевизор «Электроника-404». Традицион-



ним является подключение выхода видеосигнала БК-0010 к видеовходу телевизора через конденсатор емкостью 5—50 мкФ. Качество изображения при этом получается не слишком высоким, что связано с ограниченной (около 6 МГц) полосой пропускания видеосигнала и с тем, что видеосигнал БК-0010 имеет упрощенную форму по сравнению со стандартными. Последнее обстоятельство в случаях, когда на изображении преобладают светлые поля, часто вызывает срыв синхронизации. Увеличение уровня видеосигнала БК-0010 для получения устойчивой синхронизации приводит к дальнейшему ухудшению качества изображения.

Предлагается видеосуилитель построить по ключевой схеме (рис. 1). При этом сокращаются фронты сигналов на его выходе. Желательно также включить потенциометр для подстройки уровня видеосигнала (контрастности) в зависимости от коэффициента усиления транзистора видеосуилителя и уровня сигнала от БК-0010. На селектор синхримпульсов сигнал необходимо подать непосредственно, что обеспечивает высокую стабильность синхронизации.

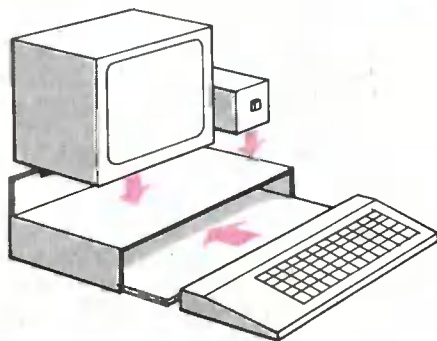
Необходимо отметить, что применение составного транзистора в видеосуилителе, вопреки ожидаемому, дало плохие результаты.

**Крепление ЭВМ пользователя.** С целью повышения электробезопасности и улучшения внешнего вида рабочего места пользователя разработана конструкция, позволяющая объединить БК-0010, блок питания и телевизор в один узел.

Применение конструкции повышает на-

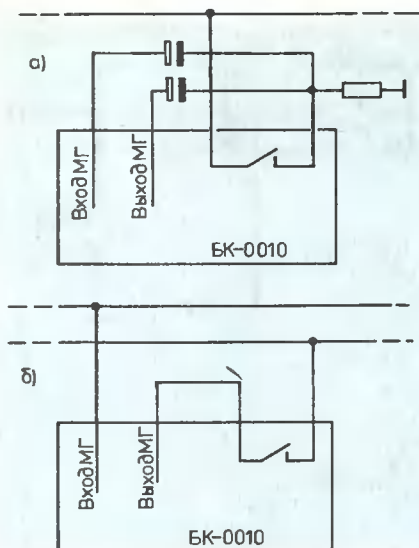
дежность рабочего места, исключая взаимные перемещения отдельных блоков и неизбежно связанное с этим дерганье разъемных соединений.

Разработаны две модификации: для образца БК-0010 с пленочной и с кнопочной клавиатурами (рис. 2).



**Магнитофонный канал.** О возможности связи отдельных БК-0010 по магнитофонному каналу уже сообщалось. Автором более детально исследованы возможности различных вариантов создания подобного канала. Наиболее целесообразными являются следующие: с помощью одного экранированного кабеля; с помощью двух экранированных кабелей.

Структурные схемы обоих вариантов показаны на рис. 3. Вариант с двумя кабелями безусловно лучше, однако в ряде случаев интерес представляет однокабельный вариант. Несмотря на то что выходы в этом



78

случае включены параллельно, можно передавать информацию между примерно пятью БК-0010.

**Система передачи изображений.** Использование параллельно с КУВТ-86 традиционной школьной доски затруднительно, так как ученики в классе часто сидят спинами друг к другу и даже к преподавателю. Поле зрения ограничивают дисплеи, перегородки и т. п.

Применение кодоскопа требует затемнения помещения, невозможна также передача динамических изображений.

А. ШМЕЛЕВ

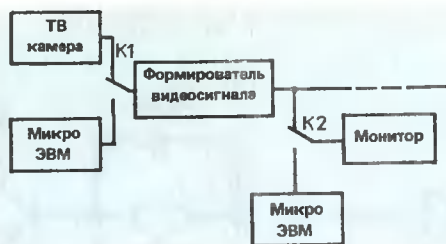
## Генератор программ-тренажеров

Система, близкая по назначению к системе «Радуга» (Информатика и образование. 1988. № 4), разработана и используется в вычислительной лаборатории Магаданского госпединститута. Ниже предлагается ее краткое описание.

В режиме диалога «ученик-компьютер» экран монитора разбит на четыре сектора (длина строки 80 символов): первый отведен для текста вопроса, задачи и т. п., второй — для текста подсказки, третий — для ответов, четвертый — для информации о правилах работы с программой, ходе выполнения заданий, количестве затраченного времени, об использовании подсказки и т. п. (пример приведен на рисунке).

Правила пользования таковы.

Клавиша F1. При ее нажатии в первом секторе появляется текст очередного зада-



Намного более хорошим решением является использование телевизионной камеры, с которой можно подать сигнал на несколько демонстрационных телевизоров, расположенных в удобных для просмотра местах.

Структурная схема системы передачи изображений ко всем рабочим местам КУВТ-86 показана на рис. 4. Она допускает следующие включения:

1. Передача на экраны сигнала от одной из ЭВМ (желательно от расположенной на рабочем месте учителя) задается положением переключателя K1. Применяется для коллективного просмотра листинга, данных или рабочей картинке, заданной учителем.
2. Передача на экраны сигнала от телевизионной камеры задается другим положением переключателя K1. Применяется для показа блок-схем программ, условий задач, фотографий и т. п.
3. Передача на экраны сигнала от ученической ЭВМ задается положением переключателя K2.

ния, т. е. задания с номером  $N+1$  (они просматриваются с начала). Если задание с таким номером выполнено, то появляется текст задания с номером  $N+2$  и т. д. до последнего. В секторе 2 появляется, например, такой текст: «ПОМНИТЕ! ЗА ИСПОЛЬЗОВАНИЕ ПОДСКАЗКИ ВАМ БУДЕТ СНИЖЕНА ОЦЕНКА»; вывод на экран монитора текста подсказки зависит от желания обучаемого. В третьем секторе предлагается меню ответов или же ввод обучаемым ответа с клавиатуры.

Клавиша F2. Действует как F1, но задания просматриваются с конца.

Клавиша F3. При ее нажатии во втором секторе появляется текст подсказки или сообщение об отсутствии таковой для данного задания.

Таким образом, зная правила пользования



ЗАДАНИЕ N 3	ПОДСКАЗКА		
Найдите корни уравнения: $2x^2+13x+6=0$	Дискриминант квадратного уравнения $Ax^2+Bx+C=0$ вычисляется по формуле $D=B^2-4AC$ . Корни уравнения вычисляются по формуле $x=(-B\pm\sqrt{D})/2A$ .		
[1с.]	[2с.]		
ВАРИАНТЫ ОТВЕТОВ	F1	F2	F3
1. $x_1=-0.5$ ; $x_2=-6$	Вопрос N+1	Вопрос N-1	Подсказка
2. $x_1=0.5$ ; $x_2=-6$	Ход выполнения задания		
3. $x_1=0.45$ ; $x_2=6$	N задания	1 2 3 4 5 ...	
4. $x_1=0.5$ ; $x_2=6$	Решение	+ - + + - ...	
Введите номер правильного ответа:	Пользование подсказкой	- + + + - ...	
[3с.]	Количество попыток	1 2 0 2 1 ...	
	Время	5 10 1 20 18 ...	
			[4с.]

программой, обучаемый может «двигаться» по заданиям и выполнять их в произвольной последовательности. Содержание того или иного сектора по желанию обучаемого (клавиша F4 с последующим указанием номера сектора) может быть выведено на экран монитора и отдельно; в этом случае используется 40-символьный режим.

Генератор данных построен так, чтобы преподаватель, не занимаясь программированием, с его помощью мог подготовить программу для проведения контрольных и самостоятельных работ. При этом достаточно иметь элементарные сведения об операционной системе и клавиатуре компьютера.

При вводе данных экран монитора разделяется на аналогичные по назначению четыре сектора, которые генератор предлагает последовательно заполнить. Длина текста вопроса, подсказки, меню ответов ограничена 440 символами, число вопросов зависит от объема текста каждого вопроса; при полной заполняемости секторов оно дости-

гает 22, чего обычно вполне достаточно. После ввода всех данных генератор подсоединяет их к первоначальной программе и записывает готовый продукт на внешний носитель.

Ясно, что и эта АОС, и, видимо, «Радуга» достаточно ограничены по своим возможностям, поскольку подготовленные с их помощью системы заданий, пусть и с различным содержанием, будут получаться внешне похожими. Важно в этой связи отметить другое: для создания программного обеспечения учебных дисциплин, в частности более совершенных генераторов, с помощью которых могли бы создаваться различные по структуре и предназначению обучающие программы, необходима координация усилий различных кафедр и вычислительных лабораторий.

Адрес для справок: 685000, г. Магадан, ул. Коммуны, 4, вычислительная лаборатория МГПИ; тел. 2-46-32.

## Рынок ЭВМ: Минрадиопром

Министерство радиопромышленности СССР — наиболее активный производитель ПЭВМ «для дома», характерные черты которых — 8-разрядный микропроцессор КР580ВМ80А и ориентация на использование в качестве внешних устройств кассетного магнитофона и телевизора. Все они имеют внешние разъемы для подключения дополнительных устройств, но только некоторые оснащены разъемами, предназначенными специально для подключения, например, принтера. Используемый совместно с ними телевизор должен иметь видеовход, а цветной — еще и RGB-вход. У подавляющего большинства телевизоров этих входов нет, но в инструкции

ях к ПЭВМ, как правило, указано, как внести в схему телевизора нужные изменения.

Сравнивая разные модели по величине ОЗУ, помните, что часть ОЗУ используется для создания изображения на экране дисплея и активное использование многоцветной многоточечной графики значительно уменьшает область хранения программ.

Быстродействие колеблется от 300 (у «Апогея») до 625 (у «Суры») тыс. операций типа «регистр-регистр» в секунду.

Прежде чем перейти к конкретным моделям, нужно упомянуть, что достоверную и полную информацию о них можно получить разве только на заводах-изготовителях; дан-

ОЗУ, К байт	«Криста»	32	«Львов-ПК01»	64	«Сура ПК 8000»	64	Апогей-БК01»	60	«Вектор-06Ц»	64	«Веста ПК 8000»	64	«Парт-нер-01.01»	64	«Искра-ту»	64
Возможность расширения ПЗУ, К байт	Да	Нет	Нет	Да	Да	Да	4	Да	Да	До 16	До 16	Да	Да	Да	20	Да
Возможность расширения ПЗУ, К байт	Да	Нет	Нет	Да	Да	Да	3	Да	Да	4/16	15	15	Да	Да	16	Да
Число цветов (градаций яркости)	2	4	4	15	15	15	3	4/16	4/16	4/16	15	15	ч/б	ч/б	16	16
Число знакомест на экране	64×25	32×24	32×24	40×24	40×24	40×24	64×25	64×25	От 32×25 до 80×25	От 32×25 до 80×25	40×24	40×24	64×25,	64×25,	64×25	64×25
Число элементов в графическом режиме	128×50	192×210	192×210	256×192	256×192	256×192	128×50	512×256, 256×256	512×256, 256×256	512×256, 256×256	256×192	256×192	512×256, 640×200	512×256, 640×200	384×256, 640×200	384×256, 640×200
Содержимое прилагаемой кассеты МК-60	Монитор	Бейсик	Бейсик	Бейсик	Бейсик	Бейсик	Монитор	Монитор	Начальный загрузчик	Начальный загрузчик	Бейсик	Бейсик	Бейсик	Бейсик	Бейсик	Бейсик
Содержимое прилагаемой кассеты МК-60	Бейсик с уроками, ассемблер, текстовый редактор, музыкальные программы, графический редактор	Тест	Тест	Тесты, игры	Тесты, игры	Тесты, игры	Бейсик, ассемблер, редактор текстов, игры, инженерные программы	Бейсик, ассемблер, редактор текстов, игры, инженерные программы	Бейсик, ассемблер, редактор текстов, игры, инженерные программы	Бейсик, ассемблер, редактор текстов, игры, инженерные программы	Бейсик	Бейсик	Бейсик	Бейсик	Бейсик	Бейсик
Завод-изготовитель	РИП, г. Муром	ЛПО им. В. И. Ленина, г. Львов	ЛПО им. В. И. Ленина, г. Львов	3-д ВЭМ, г. Пенза	3-д ВЭМ, г. Пенза	3-д ВЭМ, г. Пенза	КРЭМЗ	КРЭМЗ	КРЭМЗ	КРЭМЗ	КРЭМЗ	КРЭМЗ	КРЭМЗ	КРЭМЗ	КРЭМЗ	КРЭМЗ
Цена, руб.	510	750	750	650	650	650	650	650	750	750	950	950	650	650	500	500

*Адреса фирменных магазинов, через которые вычислительная техника*

№ 1: Москва, ул. Новокузнецкая, 17/19. № 2: Ленинград, Свердловская наб., 64. Филиал маг. № 2: Петрозаводск, ул. Ленина, 15. № 3: Ереван, 5-й Норхский массив, 13а-13б. Филиал маг. № 3: Кировскан, просп. Ленина, 34. № 5: Казань, просп. Ибрагимова, 83а. № 6: Минск, ул. Опанского, 4. № 7: Горький,

ул. Совнаркомовская, 28. № 8: Новгород, ул. Псковская, 24. № 9: Махачкала, просп. Калинина, 18. № 10: Запорожье, Центральный бульвар, 24. № 12: Ижевск, ул. Ленина, 140. «Элма»: Кишинев, Московский просп., 20.

ные, приводимые в печати, не всегда истинны. Здесь приводятся в основном данные, полученные в демонстрационном зале Минрадио-прома; однако и это не гарантирует полноты и достоверности. Например, цена «Суры ПК 8000» сейчас просто неизвестна — то ли 880 руб., то ли 1035...

Наиболее заметная особенность «Кристы» — световое перо. Оно используется для облегчения рисования на экране. К сожалению, изобразительные возможности «Кристы» ограничиваются псевдографикой.

К телевизору эта ПЭВМ подключается через обычный антенный вход; клавиатура выполнена на герконах.

У «Львова-ПК01» привлекает внимание приведенный в паспорте список ремонтных мастерских: большинство ПЭВМ чинят только на заводах-изготовителях, а эту кроме Львова еще и в Москве, Ленинграде, Алма-Ате, Ереване, Казани, Минске, Новгороде.

«Сура ПК 8000» — моноблочная конструкция (блок питания в одном корпусе с компьютером), имеет специальный разъем для принтера.

В документации на «Вектор-06Ц» не было указано, что записано на прилагаемую к нему магнитофонную кассету; этот вопрос остается открытым. Зато в ней приведены наиболее подробные и ясные указания по подключению к телевизору, не имеющему входов «видео» и «RGB». Другие особенности машины — 4-канальный звуковой синтезатор, разъемы для подключения внешнего ОЗУ (до 512К байт, используется в качестве квазидиска) и принтера.

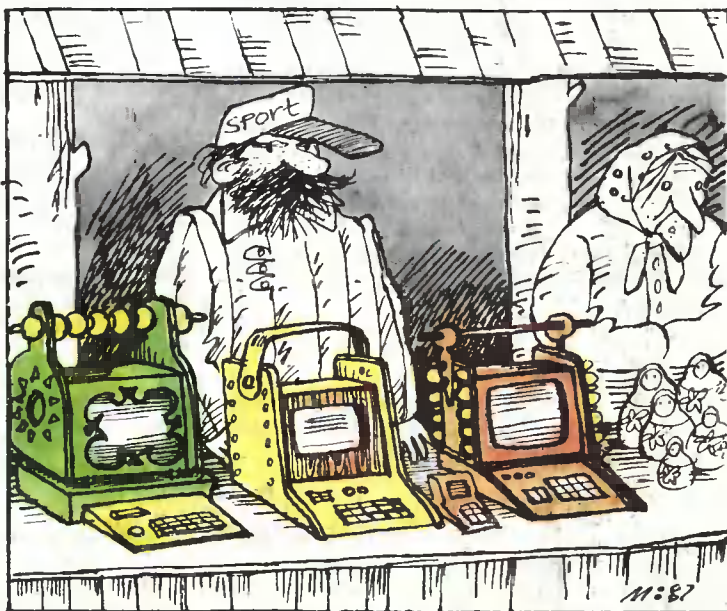
Данные моделей «Искра-1080-Тарту» и «Партнер-01.01» взяты из прессы. «Партнер» имеет моноблочную конструкцию, подключается к телевизору через антенный вход. По уверениям производителей, к нему прилагается богатое программное обеспечение, а также производятся и продаются модули расширения ОЗУ и ПЗУ, программатор ПЗУ, игровые манипуляторы, радиоизмерительные устройства и т. д.

Особняком среди этих ПЭВМ стоит «Агат». Он выпускается в трех модификациях (седьмая, восьмая, девятая), обязательно имеет дисковод (девятая модификация — даже два), правда, не лучший — на дискете 5,25 дюйма запи-



сывается только 140К байт. В комплект входит также монитор на базе телевизора «Юность Ц-404». Размер ОЗУ колеблется от 64К (у седьмой модификации) до 160К (у девятой), ПЗУ — 32К байт. Восьмая и девятая модификации комплектуются принтером Д-100. В черно-белом режиме на экран выводится 64×32 символа или 256×256 точек, в цветном — 32×32 символа, 128×128 или 64×64 точек. К компьютеру прилагаются восемь чистых дискет и две дискеты с Бейсиком и тестом. Цена «Агата» без принтера — 4800 руб. (такой комплект появляется в продаже в нефирменных магазинах), с принтером — 5400.

81



С. ЗИЛЬБЕРШТЕЙН

г. Киров

## Графика в мировых координатах

Все как-то уже привыкли, что при работе с графикой на БК-0010, «Ямахе» или «Агате» мы имеем определенное количество точек на экране по вертикали и по горизонтали и что весь графический ввод-вывод привязан к этим точкам. Но ведь такая работа естественна и удобна только для ЭВМ! Точно так же как естественны и удобны для нее машинные коды.

Для человека же наиболее естественно считать, что каждая задача имеет свою собственную область отображения, размеры и единицы измерения которой определяются самой задачей, и что эта область может быть по желанию поставлена в соответствие любому участку экрана («окну»), в частности всему экрану. При этом во всех операторах графического ввода-вывода используются координаты в терминах самой задачи. Например, если мы строим график движения, где путь изменяется от 5,1 до 38,7 км, а время от 0 до 3,6 ч, то область задачи имеет именно эти размеры, а координаты X и Y задаются соответственно в километрах и часах; и если мы определим «окно» на экране, например, в верхнем правом углу, то график должен строиться именно в этой части экрана.

Во многих развитых версиях Бейсика такие возможности имеются. При этом координаты на экране называются абсолютными, а координаты в области задачи — мировыми. Ось Y в мировых координатах, в отличие от абсолютных, направлена «естественным» образом — вверх. Например, в Бейсике IBM PC оператор

```
WINDOW (X1,Y1)-(X2,Y2)
```

определяет в терминах задачи прямоугольник области задачи (задаются верхний левый и нижний правый углы), а оператор

```
VIEW (A1,B1)-(A2,B2)
```

определяет в терминах абсолютных координат «окно» на экране (так же задаются верхний левый и нижний правый углы), на которое будет отображаться область задачи.

Изменяя значения в операторе VIEW, можно сжимать, растягивать и перемещать «окно» (вместе с изображением) по экрану. Изменяя значения в операторе WINDOW, можно изменять масштаб изображения в одном и том же «окне».

Хотя в Бейсиках БК-0010, «Ямахи» и «Агата» аналогичных операторов нет, смоделировать их работу стандартными средствами не так уж сложно. Для этого нужно задать значения величин A1, B1, A2, B2 («окно» экрана), X1, Y1, X2, Y2 (область задачи) и определить функции преобразования из мировых координат в абсолютные:

```
DEF FN X(X)=(X-X1)/K1+A1
DEF FN Y(Y)=(Y-Y1)/K2+B1
```

где K1 и K2 — постоянные коэффициенты:

$$K1=(A2-A1)/(X2-X1)$$

$$K2=(B2-B1)/(Y2-Y1)$$

После задания функций преобразования (1) в операторах PSET, PRESET и LINE (для «Агата» — PLOT) можно использовать мировые координаты в виде (FN X(X), FN Y(Y)).

Пример. Построить график функции  $Y = \sin 2x$ , где x изменяется от  $-\pi$  до  $\pi$ , отображая область задачи  $(-3.5, 1.2) - (3.5, -1.2)$  на «окно»  $(50, 100) - (200, 160)$ .

```
DATA 50,100,200,160,-3.5,1.2,3.5,-1.2
READ A1,B1,A2,B2,X1,Y1,X2,Y2
K1=(A2-A1)/(X2-X1)
K2=(B2-B1)/(Y2-Y1)
DEF FN X(X)=(X-X1)/K1+A1
DEF FN Y(Y)=(Y-Y1)/K2+B1
REM ПОСТРОЕНИЕ ГРАФИКА
PSET (FN X(-PI),FN Y(0))'НАЧАЛЬНАЯ ТОЧКА
FOR X=-PI TO PI STEP .05
LINE -(FN X(X),FN Y(SIN(2*XX)))
NEXT
```

Описанный способ достаточно прост и удобен, но обладает недостатком: если значения Y выходят за заданную область задачи (для некоторых функций, например  $Y = \lg x$ , это

будет всегда), то график функции выходит за пределы «окна». Устранить этот недостаток можно так: определить еще одну функцию

$$FNT(Y) = Y > B1 \text{ AND } Y < B2 \quad (2)$$

принимаящую значение «ложь», если Y выходит за пределы «окна», и выводить точку или отрезок только в том случае, если FNT(Y) есть «истина». Для предыдущего примера вместо оператора LINE... следовало бы записать оператор IF FNT(Y) THEN LINE...

Функции преобразования, аналогичные (1) и (2), можно определить и на Фокале БК-0010:

```
91.1 S &=(X-X1)*K1+A1;R
92.1 S &=(X-Y1)*K2+B1;R
93.1 S &=(X-B1)*(B2-X);R
```

где K1 и K2 — постоянные коэффициенты:

```
S K1=(A2-A1)/(X2-X1)
S K2=(B2-B1)/(Y2-Y1)
```

Мировые координаты в функциях Фокала FT и FV будут записываться в виде FSBR(91, X), FSBR(92, Y). Проверку на принадлежность к «окну» можно осуществлять с помощью функции FSBR(93, Y), которая должна быть неотрицательна.

Для Бейсика БК-0010 и «Ямахи», где в функциях FN допускается несколько аргу-

ментов, можно построить и более сложные функции преобразования, например функции, позволяющие кроме сжатия, растяжения и переноса изображения поворачивать его на заданный угол A вокруг левого угла «окна».

Функции преобразования в этом случае выглядят так:

```
DEF FN X(X,Y)=(X-X1)*K1+(Y-Y1)*K2)*K5+A1
DEF FN Y(X,Y)=-(X-X1)*K3+(Y-Y1)*K4+B1
DEF FNT(X,Y)=X>X1 AND X<X2
AND Y>Y1 AND Y<Y2
```

где K1—K5 — постоянные коэффициенты:

```
K1=(A2-A1)/(X2-X1)*COS(A)
K2=(B2-B1)/(Y2-Y1)*SIN(A)
K3=(A2-A1)/(X2-X1)*SIN(A)
K4=(B2-B1)/(Y2-Y1)*COS(A)
K5=(PEEK(32)AND1XOR1)+1)*0.72
```

Коэффициент K5 введен для устранения неравномерности графики по осям в БК-0010; в «Ямахе» он равен 1. Функция FNT контролирует нахождение точки в «окне» не только по Y, но и по X.

В заключение следует отметить, что работа с графикой в мировых координатах кроме простоты использования и возможности управления изображением имеет еще одно достоинство: для переноса программы на ЭВМ с другими характеристиками дисплея нужно лишь изменить в программе размеры «окна».

## Отзовитесь, коллеги!

Хочу сообщить, что мной разработан для БК-0010 транслятор языка Си — вопреки сложившемуся мнению, что трансляторы языков такого класса могут быть реализованы лишь на гораздо более мощных компьютерах.

Транслятор воспринимает несколько урезанный вариант языка, вполне достаточный, однако, чтобы писать на нем достаточно серьезные программы. Результатом работы являются не машинные коды, а текст на языке ассемблера МИКРО II. Это позволяет делать включения текста на ассемблере в текст программы на Си, однако, с другой стороны, причиняет некоторые неудобства в работе.

Разработка транслятора, на мой взгляд, открывает новый этап в программировании на БК. Становится возможным программировать на более высоком профессиональном уровне без применения кросс-ЭВМ, использовать один из наиболее популярных в мире

языков программирования. Да и в обучении программированию можно будет перейти от убогих Бейсика и Фокала к Си — языку современному, реализующему практически все конструкции структурного программирования, языку для профессионального программирования и реализованному практически на всех ЭВМ более высокого класса. В настоящее время я планирую работу по улучшению транслятора — генерацию непосредственно машинных кодов, разработку Турбо-версии, пополнение библиотеки подпрограмм.

Я слышал, что, кажется, в Саратове ведутся подобные разработки (трансляторы Си, Паскаля). Нет ли у вас какой-либо информации об этом? В любом случае, не могли бы вы сообщить мне адреса владельцев БК из Саратова, известные вам?

С уважением  
А. В. Цаплев

199048, Ленинград, а/я 122

# ЧТО МОЖЕТ ЭВМ?

## Мозг для кондиционера

В один из жарких августовских дней, когда температура воздуха превысила температуру тела участников конференции в Чикаго, на трибуну поднялся человек, представивший разработанный его фирмой «думающий» кондиционер. Установив такой прибор в квартире, вы можете смело оставить ее на любое время. К вашему приходу воздух в комнате будет приведен до нужной кондиции без лишнего затрат энергии.

Экономичный режим работы кондиционера обеспечивает встроенный микропроцессор, с помощью которого вы можете установить четыре различные температурные программы на рабочие дни недели и две — на выходные.

Для тех, у кого кондиционер уже есть, разработчики предусмотрели небольшой прибор, выполненный в виде вставляемой прямо в розетку коробочки, в которую можно включить имеющийся бытовой прибор. Комнатное устройство позволит повысить «интеллект» вашего кондиционера до уровня его программируемого брата.

## Электронный хамелеон

Что общего у профессий летчика-испытателя, литейщика и сварщика? Резкая и частая смена освещенности рабочего поля. Такая, что даже предусмотренный природой механизм защиты глаз не справляется с нагрузкой.

Самым простым и дешевым способом создания человеку комфортных оптических условий являются светофильтры различной плотности. У сварщиков это темный, встроенный в маску ку-

сочек стекла. У отдыхающих на пляже дымчатые очки. Но и маска, и очки начинают мешать, как только освещенность уменьшается.

Тем, кто вынужден носить очки постоянно, химики предложили стекло, коэффициент затемнения которого зависит от освещенности. На солнце очки, сделанные из такого стекла, через несколько минут темнеют, а в комнате вновь становятся почти прозрачными.

Окно из такого стекла днем будет поддерживать в комнате примерно постоянную освещенность.

Однако было бы полезнее, если бы прозрачность оконного стекла управлялась не светом, а по сигналу компьютера. Тогда в аудитории перед демонстрацией опыта или в кинозале перед сеансом можно было бы за доли секунды установить нужный уровень освещенности, не пользуясь шторами, тем более что такие стекла уже сравнительно широко вошли в наш быт в виде жидкокристаллических дисплеев, используемых в наручных часах, калькуляторах и персональных компьютерах.

Американские компьютерщики решили изготовить управляемую напряжением жидкокристаллическую панель большого размера и не прогадали. Их продукция, несмотря на дороговизну, пользуется спросом. И теперь в домах владельцев таких стекол вполне можно услышать: «Дорогая, включи, пожалуйста, окно. Уже, должно быть, утро!»

## Интерактивный... телевизор

С появлением на рынке новых бытовых видеосистем, позволяющих не только наблюдать за происходящим на экране, но и активно воздействовать на него, недозволенных сюжетными завязками телефильмов поубавится.

Видеосистема, основу которой составляет высококачественный телевизор со встроенным компьютером, может получать информацию как от кабельной системы или антенны, так и со стандартной видеокассеты или видеодиска. В зависимости от содержания источника информации вы можете позволить себе стать управляющим крупного предприятия, дела которого в виде довольно реалистичного фильма будут разворачиваться перед вами на экране телевизора, или пилотом

межгалактического корабля, ведущим жестокие бои с космической нечистью, или «кабинетным» мыщиком, распутывающим одно преступление за другим. Причем каждый раз, в зависимости от уровня вашей квалификации и принятых на предыдущих шагах решений, система будет предлагать все новые и новые жизненные ситуации.

## Компьютер конструирует улыбку

Специалисты американской фирмы «Дата транслейшн» создали сопроцессор DT2861 для IBM PC AT, который будет хорошим помощником фотографу и его клиентам.

В каком бы настроении вы не находились, мощная графическая плата с сопроцессором DT2861 сумеет придать вашему облику наиболее естественный вид. Более того, она же поможет подобрать прическу, даже если у вас нет ни одного волоса, сконструирует костюм, снимет или наденет на вас очки, предложит усы и бороду нескольких фасонов и даже, если это необходимо, изменит ваш возраст.

Сопроцессор незаменим не только для полицейских, которые по самым незначительным признакам смогут построить достоверный портрет преступника, но и для художников, модельеров, парикмахеров и даже режиссеров, которые всего за несколько минут смогут увидеть своего героя в различных настроениях, позах и т. д.

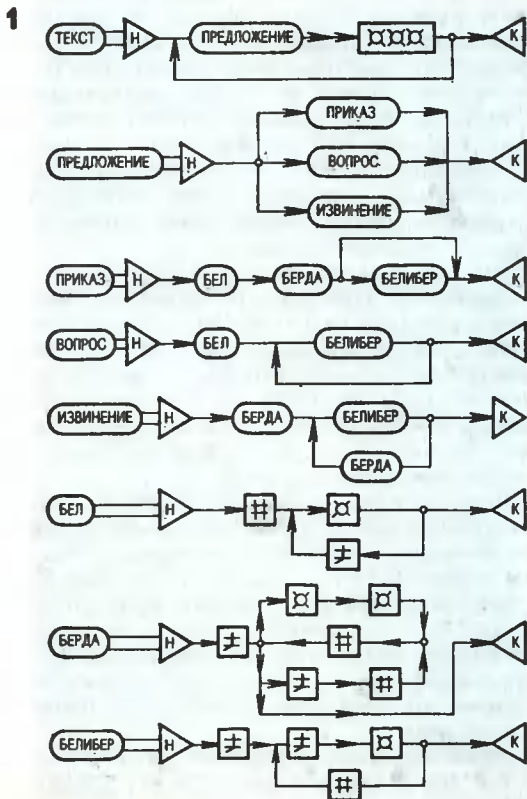
Все преобразования осуществляет комплекс программных средств, поставляемых с сопроцессором. Компьютер, укомплектованный сопроцессором, может видоизменять одновременно четыре цветных изображения или демонстрировать на экране сразу 16 уже обработанных портретов, хранящихся в его памяти одновременно. Для формирования каждого портрета отводится поле размером 512×512 пиксел, что обеспечивает высокое качество цветного изображения. Плата может работать с IBM-совместимыми ЭВМ и принимает информацию от таких источников видеосигналов, как оптический сканер, посточно-считывающая камера, электронный сканирующий микроскоп, и от обычной видеокамеры, которой комплектуются бытовые видеомэгафоны.

Л. САНЖАРОВ, А. ФИНЬКОВ  
Измаильский педагогический институт

## Квасиязык на ЭВМ в учебных целях

В книге Г. А. Звенигородского «Первые уроки программирования» (М.: Наука, 1985. С. 103—104) помещена задача об условном белибердинском языке.

Алфавит белибердинского языка состоит из трех символов:  $\ddagger$ ,  $\boxtimes$  и  $\#$ . На основе анализа многочисленных надписей ученым удалось установить, что синтаксис языка описывается диаграммой (рис. 1).



Смысл предложений языка не выяснен. На экспертизу в белибердоведческую лабораторию поступили две новые надписи (рис. 2). Предполагается, что одна из них содержит подлинный текст, а другая является подделкой. Постарайся провести экспертизу и обнаружить подделку, если она есть. Проведи синтаксический разбор подлинного текста и определи, сколько в нем приказов, сколько вопросов и сколько извинений.

Суть задачи в следующем. Белибердинский язык, относящийся к числу квазиязыков, имеет письменность, представляющую собой различные комбинации трех знаков:  $\ddagger$ ,  $\boxtimes$ ,  $\#$ . Структура языка подчиняется правилам, которые представлены 8 синтаксическими диаграммами (алгоритмами), определяющими структуру текста, вид предложения, структуру предложения, структуру каждого элемента предложения. Особенность языка: в каждом типе предложений, в каждом его элементе могут быть циклы, что значительно усложняет и чтение, и письмо.

Нас заинтересовала предложенная Г. А. Звенигородским задача. Во-первых, белибердинский язык может рассматриваться как своеобразная модель языка вообще, а на модели всегда удобнее и нагляднее можно показать черты естественных языков при их изучении. Во-вторых, показалось интересным реализовать предложенную в условии задачи структуру языка (синтаксические диаграммы — алгоритмы) на ЭВМ. Нужно отметить, что предложенная Г. А. Звенигородским наполовину шуточная задача оказалась, к чести автора, глубокой и интересной. Она позволяет иллюстрировать довольно серьезные в теоретическом плане положения. И этим хочется сказать еще одно доброе и вполне заслуженное слово в память о так рано ушедшем ученом.

- а) 井 ㄣ ㄣ ㄣ ㄣ ㄣ ㄣ 井 ㄣ ㄣ ㄣ ㄣ 井 ㄣ ㄣ ㄣ ㄣ  
 ㄣ ㄣ ㄣ ㄣ 井 ㄣ ㄣ ㄣ ㄣ 井 井 ㄣ ㄣ ㄣ ㄣ 井 ㄣ 井 井 ㄣ ㄣ  
 ㄣ ㄣ 井 井 ㄣ ㄣ 井 ㄣ 井 ㄣ 井 ㄣ ㄣ ㄣ ㄣ 井 ㄣ ㄣ ㄣ ㄣ ㄣ ㄣ  
 ㄣ ㄣ 井 井 ㄣ ㄣ 井 ㄣ ㄣ ㄣ ㄣ
- б) 井 ㄣ ㄣ ㄣ 井 ㄣ ㄣ ㄣ ㄣ 井 ㄣ ㄣ ㄣ ㄣ ㄣ 井 井 ㄣ ㄣ  
 ㄣ ㄣ ㄣ 井 ㄣ ㄣ ㄣ ㄣ ㄣ 井 ㄣ 井 ㄣ ㄣ ㄣ ㄣ ㄣ 井  
 ㄣ ㄣ ㄣ 井 井 ㄣ ㄣ ㄣ ㄣ ㄣ ㄣ ㄣ ㄣ 井 ㄣ ㄣ ㄣ ㄣ ㄣ 井  
 ㄣ ㄣ ㄣ

Анализируя условия задачи, перед студентами можно поставить следующие вопросы.

Сравнивая особенности языковой модели, предложенной в задаче, и естественный язык (например, русский), какой метод исследования языков мы применяем? Почему?

В русском языке выделяются повествовательные, вопросительные и побудительные предложения, а в белибердинском языке (если использовать нашу терминологию) — вопросительные, побудительные и извинительные. На каком языке общаться легче и почему? Какой язык передает больше информации и почему?

В чем особенность системы письма в русском (английском, французском и т. д.) языке по сравнению с белибердинским? Исходя из этой особенности, какой синхронный срез письма (русский, белибердинский) является более древним? Почему?

Как формируется каждый элемент белибердинского языка? Если нарушается порядок следования графем, характер «окружения» составных элементов графем, сохраняется ли значение каждого элемента? Для решения этих вопросов какой метод изучения языков можно использовать?

Учитывая особенности предложений по цели высказывания в белибердинском языке, что мы можем сказать о характере говорящего на нем народа?

Чем являются по своей сути белибердинские знаки ㄣ, 井, ㄣ? Создает ли столь ограниченное число графем трудности для письма в белибердинском языке? Почему? Можно ли передать русский текст этими графемами, если принять за условие, что их только три и что они формируют элементы текста по тем же алгоритмам? Может быть, у вас возникли фантастические предложения о переводе русской графики на белибердинскую основу? Не могли бы вы подробнее рассказать о такой ситуации?

Эти вопросы и ответы студентов позволяют в весьма занимательной и необычной форме

выяснить такие общезыковедческие положения, как методы изучения языков, содержание и форма в языке, язык и народ, особенности графики и орфографии, вопросы их усовершенствования. Причем идут не просто общие рассуждения на предложенную тему, а с опорой на сопоставление по крайней мере двух языков.

Может возникнуть вопрос, зачем нужен квазиязык, если беседы и сопоставления можно проводить на материале естественных языков. Да, конечно, можно. Но, во-первых, трудно найти студенческую аудиторию, особенно на первом курсе, которая одинаково хорошо владеет хотя бы двумя естественными языками: родным и каким-то другим. Во-вторых, квазиязык, уводя от смысла высказывания, заостряет внимание на структуре, форме, делает их очень наглядными. В-третьих, есть авторитетнейший прецедент: академик Л. В. Щерба понятия о грамматическом значении и грамматической форме показывал на широко теперь известной квазифразе «Глока куздра штеко будланула бокра и курдячит бокренка».

После беседы по вопросам мы приступаем к следующему заданию — расшифровке текстов и установлению их истинности или подделки. Эта работа тоже имеет лингвообразовательное значение: кто знает, может быть, нашим студентам придется в жизни заниматься расшифровкой настоящих непонятных текстов. Пусть знают хотя бы один из возможных для этого путей.

Четкость алгоритмической сущности задачи привела нас к мысли использовать для ее решения компьютер, что представляло исследовательский интерес и показывало применение ЭВМ в лингвистике. Кроме того, мы получили еще одну возможность рассмотреть чисто лингвистический вопрос: рассказать студентам о понятии порождающей грамматики, проиллюстрировав ее принципы в действии.

Наша программа написана на Бейсике. В ней предусматривается машинная провер-



ка проделанной студентами работы по расшифровке текстов: студенты вводили в ЭВМ предложенные Г. А. Звенигородским тексты, машина анализировала их и указывала, какой текст является верным, а какой поддельным. В первом случае компьютер производит и грамматический анализ текста: выдает сообщение о том, какие типы предложений и в каком количестве представлены в тексте. Во втором случае, получив сообщение, что текст поддельный, студент набирает его по предложениям, машина оценивает истинность каждого предложения и указывает те из них, где нарушены законы квазиязыка.

Вторая практическая задача состояла в том, чтобы машина по запросу студента могла «порождать» текст на белибердинском языке. Изучая метод лингвистического анализа по непосредственно составляющим, мы говорили, что он дает возможность выделить в языке основные элементы и правила их соединения. Если их заложить в ЭВМ, то из этих элементов и по этим правилам компьютер сможет конструировать текст. Программа предусматривает конструирование текста в пределах до 1000 знаков, выдачу его на экран, а при необходимости и на принтер, грамматический анализ этого текста по типам предложений. Порождаемые машиной тексты строго соответствуют алгоритмам квазиязыка, и, как правило, один текст не повторяет другой.

Блоки программы:

«служебная» часть: выходные данные про-

граммы, ввод информации о студенте, изложение основных правил белибердинского языка, общее заключение о результатах работы студента;

ввод студентами текста и проверка его компьютером;

печать введенного текста на экране и принтере;

составление текста машиной по просьбе студента.

Программа составлена для машины ДВК-2М, не имеющей, как известно, графического дисплея. Поэтому синтаксические схемы, указанные Г. А. Звенигородским, на нашем экране выглядят иначе.

Программа получилась объемной, поэтому мы не приводим ее листинга.

В заключение отметим, что использование данной задачи на занятиях по языкознанию оказалось весьма эффективным, так как показало студентам возможности компьютера при решении лингвистических задач. В интересной форме были продемонстрированы явления языка, которые при традиционном обучении воспринимались обычно как нечто отвлеченное и умозрительное. Задачу, не вдаваясь в специальный, лингвистический анализ, можно с успехом использовать и на уроках информатики, демонстрируя ее возможности при изучении разных дисциплин.

Со всеми, кто заинтересуется нашей работой, мы готовы установить контакт и рассказать на более предметном уровне о принципах составления описанной программы.

А. СИМОНОВ, И. СЫСОЕВ, Ю. ШАХОВА

## Программа-тренажер по астрономии

Профессиональная подготовка студентов в вузе в настоящее время приобретает особую актуальность. Одним из существенных компонентов этой подготовки является изучение студентами теоретических курсов. Однако традиционная вузовская методика не обеспечивает высоких результатов обучения, так как имеет следующие недостатки: стихийно складывается процесс усвоения знаний; традиционная методика вузовской лекции в значительной мере обрекает студента на пассивность; структура учебников построена фрагментарным способом. Увеличение объема самостоятельной работы студентов и сокращение лекционных курсов при существующей методике не решают проблему, а лишь усугубляют ее. Кроме того, в процессе практи-

ческой реализации этого направления наметился подход, при котором не особенности процесса учения стали определять требования к техническим средствам, а, наоборот, процесс учения стали приспособлять к возможностям технических устройств, что, на наш взгляд, не является правильным.

Возможности современных ЭВМ позволяют полнее использовать достижения психолого-педагогических наук, в частности идеи программированного обучения (П. Я. Гальперин, Н. Ф. Талызина). В разработанной авторами программе сделана попытка реализовать некоторые из них.

Программа-тренажер по астрономии написана на Бейсике для ЭВМ «Ямаха» по

теме «Основные элементы небесной сферы и ее суточное вращение» раздела «Сферическая астрономия». Она включает инструкцию для пользователя, средства диалогового взаимодействия с ЭВМ, средства контроля усвоения и оценки знаний студентов.

Выбор темы для этой программы обусловлен следующими психолого-педагогическими факторами.

Программа-тренажер охватывает материал примерно двух лекций раздела «Сферическая астрономия». На этих лекциях, как правило, вводятся основные функциональные понятия науки, на основе которых строится дальнейшее изучение этого и, частично, других разделов. Кроме того, вводится большое количество — около тридцати — новых понятий и терминов, что затрудняет дальнейшее усвоение студентами лекционного курса. В итоге происходит снижение качества лекций и возрастание трудностей в работе студентов с учебной литературой; как следствие — потеря интереса к предмету. На последующих лекциях вводятся два-три новых понятия, но их качественная обработка возможна при условии хорошего усвоения материала первых двух лекций. При традиционной методике обучения из-за большой частоты лекций это осуществить трудно.

Для быстрой, успешной обработки большого количества новых понятий и терминов и создана данная программа-тренажер. Обучение по ней строится в трех вариантах. Это обусловлено тем, что для усвоения знаний обучаемому необходимо поэтапно отработать действия в определенной последовательности, а именно в материализованной, речевой и умственной форме (П. Я. Гальперин). Программа-тренажер обеспечивает работу студента последовательно на каждом этапе и состоит соответственно из трех подпрограмм.

Отработка действий в материализованной форме осуществляется в процессе работы студентов с рисунками на экране дисплея. Работа организована следующим образом: на периферии помещаются стилизованные изображения элементов небесной сферы, а задача студента заключается в перенесении элементов в центр небесной сферы путем перемещения указателя в область экрана,

занимаемую данным элементом. В основу этой работы положена идея о строгой последовательности (а не фрагментарности) построения основных элементов небесной сферы.

Во время отработки материализованных действий на экране дисплея высвечивается название элемента небесной сферы. Запрограммированы пооперационный контроль на основе обратной связи и подсказка. Если студент нарушил последовательность построения, на экране появляется надпись: «Вы ошиблись. Следующий элемент небесной сферы...» (называется элемент, например небесный экватор).

Для отработки речевых действий из программы исключается подсказка. В этом случае студенту разрешается самостоятельно, подумав (речевое действие), исправить ошибку.

Для отработки умственных действий из программы исключается также и пооперационный контроль. При работе с этой подпрограммой учитываются все ошибки и выставляется оценка за работу по построению. В этом случае построение осуществляется в умственной форме, а экран дисплея служит только для фиксации того, что построено студентом в уме. Поэтому и контроль осуществляется по конечному результату. Этой же цели служат и контрольные вопросы.

Таким образом, реализация некоторых идей программированного обучения, а также усиление мотивации обучения за счет вкрапления игровых ситуаций привели авторов статьи к мысли о возможности организации самоподготовки студентов с помощью компьютера.

Программа используется при обучении студентов 4—5-х курсов физико-математического факультета. На эти занятия студенты идут с интересом, работают с компьютером охотно, усваивают знания легко, отвечают компьютеру свободно, раскованно. Следующую лекцию, на которой новые понятия уже начинают «работать», студенты слушают с пониманием, лекция вызывает интерес, превращается в беседу.

Авторами в течение двух лет проводился эксперимент, в котором сравнивались результаты обучения студентов традиционным ме-

Годы	Метод обучения	Всего студентов	Полученные оценки			
			5	4	3	2
1986—1987	Традиционный	126	8	12	83	21
		128	30	57	40	1
1987—1988	С применением программы-тренажера					

тодом и с использованием программы-тренажера.

Результаты приведены в таблице.

Данная программа-тренажер может быть с успехом использована в школе на кружковых и факультативных занятиях; на уроках при изучении темы «Основные элементы

небесной сферы», на которую отводится небольшое количество часов.

Желающие ознакомиться с программой могут написать по адресу: 417001, г. Уральск, пр. Ленина, 162; педагогический институт им. А. С. Пушкина, кафедра информатики и вычислительной техники.

А. КОГАН, СПТУ-31, г. Керчь

## Отчего кошку назвали кошкой?

В упражнении 10 (§ 3, ч. 1) требуется объяснить условие *кошка*-«кошка» и ответить на вопрос: «Что является именем, что — значением величины?».

Предлагаю комментировать это упражнение монгольской народной сказкой в переводе С. Я. Маршака\*, название которой (см. заголовков) полностью соответствует содержанию упражнения:

У старика и старухи  
Был котенок черноухий,  
Черноухий  
И белощекий,  
Белобрюхий  
И чернобокий.

Стали думать старик со старухой:  
— Подрастает наш черноухий.  
Мы вскормили его и вспоили.  
Только дать ему имя забыли.

Назовем черноухого  
«Тучей» —  
Пусть он будет большой  
И могучий.  
Выше дерева.  
Больше дома.  
Пусть мурлычет он громче грома!



— Нет,— сказала, подумав, старуха,—  
Туча легче гусяного пуха.  
Гонит ветер огромные тучи,  
Собирает их в серые кучи.

Свищет ветер  
Протяжно и звонко.  
Не назвать ли нам «Ветром»  
Котенка?

— Нет, старуха,—  
Старик отвечает,—  
Ветер только деревья качает,  
А стена остается в покое.  
Не назвать ли котенка  
«Стеною»?

Старику отвечает старуха:  
— Ты лишился на старости слуха!  
Вот прислушайся вместе со мною:  
Слышишь, мышка шуршит за стеною?  
Точит дерево мышка-воришка...  
Не назвать ли нам кошку — «Мышка»?

— Нет, старуха,—  
Старик отвечает,—  
Кошка мышку со шкуркой съедает.  
Значит, кошка  
Сильнее немножко!  
Не назвать ли нам кошку кошкой?..

Обратив внимание учащихся на тип рассматриваемой величины (переменная), можно истолковать рассматриваемое равенство как результат последовательного выполнения следующих команд присваивания:

кошка:=«туча»  
кошка:=«ветер»  
кошка:=«стена»  
кошка:=«мышка»  
кошка:=«кошка»

Использование такого методического приема вызывает живой интерес у учащихся и способствует сознательному усвоению учебного материала.

\* Маршак С. Я. Сказки, песни, загадки. М.: Детская литература, 1966. Т. 2. С. 173.

Л. КУЛОВА, Ю. МОВСЕСЯН  
Орджоникидзе

## Как учить информатике сегодня?

По приближенным оценкам специалистов, в народном хозяйстве страны уже сейчас следовало бы иметь 20—25 млн. персональных ЭВМ, а наша промышленность планирует на будущую пятилетку выпуск всего 1,5 млн. ПЭВМ. По заключению ученых из Института философии АН УССР, по готовности к информатизации наша страна занимает лишь 62-е место в мире. Наш массовый потребитель не подготовлен к применению новых средств, не видит возможностей их использования для решения производственных, научных и других задач, хотя признаки информатизированного общества известны и успешно реализуются в развитых капиталистических странах. Это и разветвленная сеть коммуникаций, доступная каждому гражданину, и компьютеризация всех сторон жизни. Это, наконец, право и реальная возможность любому человеку незамедлительно получать из общедоступных банков данных любую информацию, кроме той ее незначительной части, которая составляет государственную, коммерческую или частную тайну.

Но информатизация и компьютеризация не самоцель и уж тем более не дань современности и моде. О том, какие ощутимые плоды они приносят обществу, говорят следующие цифры. В 1985 г. доля программного продукта в общем объеме национального дохода США составила 8 %. Это столько же, сколько принесли стране сельское хозяйство, автостроение и добывающая промышленность вместе взятые. Вклад же индустрии связи составил 3,2 %. В Советском Союзе связь дает менее 1 % валового национального дохода, а доля программного продукта еще ниже.

Не лучше обстоят у нас дела и в сфере образования. Предполагалось в течение 1986—1990 гг. выпустить 500 тыс. персональных микро-ЭВМ, не уступающих лучшим мировым образцам. Совершенно очевидно, что эти планы в количественном отношении не будут выполнены. Что же касается качества школьных ЭВМ, то здесь разговор особый. Ведь пока школьные компьютеры представляют собой «букет недостатков»: они и дороги, и ненадежны, и объем памяти



у них мал, и не предусмотрено сопряжение с набором периферийных устройств.

Несмотря на удручающую реальность, представим себе, что завтра (или в начале следующей пятилетки) мы будем обеспечены прекрасными школьными ПЭВМ, как нам это обещал председатель Госплана СССР тов. Ю. Маслюков на сессии Верховного Совета СССР. Тогда мы окажемся лицом к лицу с другой проблемой: как их эффективно использовать.

Так как же все-таки учить информатике сегодня?

Считается, что основы компьютерной грамотности закладываются в школах и ПТУ в курсе, введенном с 1985 г. Парадоксально, но факт, что в большинстве учебных заведений он читается в «безмашинном варианте», т. е. без практического выхода на ЭВМ. «Безмашинный вариант» реализуется либо в случае отсутствия ЭВМ, либо в случае неумения обращаться с ними преподавателей, либо в случае невозможности ремонта вышедшей из строя техники. Последнее, на наш взгляд, наиболее социально опасно, так как порождает неверие в возможности ВТ, дискредитирует саму идею компьютеризации. Курс ОИВТ, изложенный без практического выхода на машины, воспринимается учащимися как бесполезный, сугубо теоретический предмет и вызывает у них лишь чувство антипатии.

В соответствии с постановлением ЦК КПСС и Совета Министров СССР от 29.03.85 г. вычислительная техника, поступающая в систему народного образования, концентрируется в межшкольных УПК, районных кабинетах и других подобных учреждениях, где ее легко ремонтировать и обеспечивать ее эффективную загрузку. Такую стратегию в распределении ВТ следует, на наш взгляд, считать рациональной.

В рамках реализации стратегии комплексного использования ВТ при Министерстве народного образования Северо-Осетинской АССР в феврале 1989 г. в Орджоникидзе был открыт экспериментальный Политехнический центр. Создание его происходило при активном и непосредственном участии баз-

вого предприятия — научно-производственного объединения «Полимер». На сегодняшний день центр уже обладает солидной для просвещенческого учреждения материальной базой: тремя классами по 26 учебных мест, оснащенными КУВТ-86; двумя классами по 26 мест, оснащенными КУВТ «Корвет»; одним классом, оснащенным КУВТ «Ямаха», на 26 ПЭВМ и машинным залом, в котором размещены 18 ПЭВМ отечественного и зарубежного производства различных моделей. Кроме этого, в центре имеется лаборатория микропроцессорных устройств и микро-ЭВМ; лаборатория станков с ЧПУ и роботов; цех металлообработки, содержащий 24 единицы оборудования с электронным управлением.

На базе центра в этом учебном году осваивают курс ОИВТ около 3 тыс. учащихся ПТУ, более полутора тысяч десятиклассников, более 500 студентов медицинского института и 600 инженерно-технических работников НПО «Полимер». В ближайшем будущем центром будет осуществляться подготовка и переподготовка рабочих по обслуживанию и ремонту новой техники с электронным управлением. Кроме этого, с 1991 г. центр будет выпускать товары народного потребления и продукцию металлообработки по заказам базового предприятия общим объемом на 1 млн. рублей.

Но Политехнический центр, по нашему мнению, не просто коллекция сложной техники, не просто редкая по обеспеченности база для народного образования. Это прежде всего люди. Коллектив единомышленников, энтузиастов. Ведь хорошо известно, что инициативу невозможно «возложить» ни на отдельного человека, ни на коллектив, ее проявляют только самостоятельно. Вот почему планы нашего центра — это нечто гораздо большее, чем круг обязанностей, предусмотренных должностными инструкциями.

С сентября этого года в качестве эксперимента центр начал обучение ОИВТ в 22 четвертых классах общеобразовательных школ,



начал с игр на компьютерах, постоянного общения с машинами, с постепенного введения ЭВМ в учебный процесс. А к моменту окончания средней школы будущие десятиклассники вместе с аттестатом зрелости получают диплом техника-программиста.

В качестве операторов в центре работают только студенты вечернего и заочного факультетов Северо-Осетинского государственного университета, обучающиеся различным специальностям, в том числе и гуманитарного цикла. По окончании университета будущие преподаватели географии, истории, иностранного языка приобретут навыки свободного общения с ЭВМ, осознают возможности их практического использования в школьном учебном процессе. В соответствии с приобретенной в университете специальностью они будут распределены в спецшколы и получат в свое распоряжение дисплейные классы. Приход их в эти школы обеспечит качественно новую стадию в компьютеризации, так как позволит соединить опыт педагогов-мастеров по предметам со свободным владением ЭВМ молодыми преподавателями.

С начала нового учебного года центр взял на себя обязанность обслуживать малокомплектные и небольшие сельские школы в отдаленных горных районах республики. Подготовлен к выезду автобус «Икарус-260», в салоне которого размещены 26 ПЭВМ «Правец». Проводят занятия с сельскими детьми студенты кафедры прикладной математики университета под руководством преподавателя-консультанта центра.

Министерством народного образования СО АССР совместно с ВНИИ профтехобразования АПН СССР на базе Политехнического центра с использованием его специалистов и техники начат широкомасштабный эксперимент по определению и повышению уровня педагогического мастерства учителей средних школ и преподавателей ПТУ республики. Для участия в нем и для решения





сложных психолого-педагогических задач, встающих перед центром, привлечены веду-

щие специалисты средней и высшей школы. При центре создан научно-методический совет, в состав которого вошли профессор и доценты Северо-Осетинского государственного университета, Северо-Кавказского горно-металлургического института, медицинского института, преподаватели-мастера школ и ПТУ нашей республики.

В этой небольшой по объему статье невозможно было даже кратко изложить все инициативы нашего центра, рассказать о всех наших планах. Может быть, наши убеждения не являются бесспорными, но если мы хоть малой частью наших начинаний ответили на вопрос, вынесенный в заголовок статьи, мы будем считать свою задачу выполненной.

Адрес для переписки: 362045, г. Орджоникидзе, ул. А. Кесаева, 12-В, Политехнический центр. Тел. 7-94-14.

92

## Фотокомбайн

У хорошего фоторепортера всегда под рукой несколько камер, сменных объективов, светофильтров и масса других мелочей, позволяющих быстро и наилучшим образом снимать интересные и неожиданные сюжеты в самых разных условиях, поэтому работа фотокорреспондента часто больше похожа на труд носильщика, чем на творчество художника.

Значительно облегчило участь профессиональных фотографов появление камер со встроенным микропроцессором, который не только помогает определять и устанавливать экспозицию, но и держит изображение в фокусе, помещает отснятый материал и сам с помощью встроенной фотовспышки регулирует освещенность.

Но и кремниевое чудо не сумело справиться со всем разнообразием проблем, с которыми фоторепортер сталкивается ежедневно. Поэтому одни фирмы начали выпускать камеры с микропроцессором для спортивных съемок, другие — для съемок портретов, третьи — для съемок репродукций. И вновь сумка фотографа стала пухнуть.

Изысканное и простое решение этой проблемы предложила одна из компаний, производящих фотоаппараты. В их продукцию микропроцессор не встраивается жестко, а вставляется с помощью

разъема в гнездо на задней стенке фотокамеры. Такой фотоаппарат комплектуется набором микропроцессоров, каждый из которых ориентирован на свой вид съемки. Горсть компактных и легких чипов в комплекте с подобной камерой заменяет несколько изысканных фотоаппаратов для специальных видов съемки. Кроме того, у потенциальных покупателей пропадает страх перед поломкой нежной электронной начинки. Ведь сменить ее не сложнее, чем батарейку.

**ЧТО?**  
МОЖЕТ?  
**ЭВМ**

## Как оживить идею

Опытный инженер по чертежу может понять, для чего предназначено сконструированное устройство. Но о работе сложного механизма даже автор проекта вряд ли сможет сказать что-либо определенное до тех пор, пока не будет собран опытный образец. Только воплощенная «в металле» конструкция позволяет отработать все идеи своих создателей, определить надежность, проч-

ность и другие характеристики.

А нельзя ли всю эту работу проделать в памяти компьютера? «Можно», — считают разработчики моделирующей программы «ЗД Анимейшен» из США. С помощью предлагаемого ими инструмента вы можете не только спроектировать новую конструкцию и начертить ее на плоттере, но и просчитать ее прочностные характеристики во всех допустимых режимах работы, оптимизировать размеры и вес, уменьшая толщину материала там, где это возможно, и даже увидеть задуманную вами машину в действии на экране дисплея в трехмерном изображении. Моделирующая программа избавит вас от многочисленных доработок и переделок опытного образца и вообще необходимости его изготовления. С ее помощью вы можете провести полный цикл испытаний новой конструкции, проверив ее работу даже в таких режимах, которые на реальном изделии недостижимы.

Хорошая моделирующая система может «оживить» на экране компьютера даже не существующие в реальном мире объекты, такие, как математическая функция или система массового обслуживания. Моделирование и визуализация поведения таких объектов позволяет порой найти правильное решение значительно быстрее, чем классические аналитические методы.

В. ЗЕЛЕНИН

ЛГПИ им. А. И. Герцена

## Метод проектирования ППС по структурированным образцам на языке MSX-Бейсик

В педагогических вузах страны началось обучение студентов по специальности «Учитель информатики и вычислительной техники». В учебные планы подготовки по ней включена дисциплина «Использование вычислительной техники в учебном процессе», в которой имеется раздел «ЭВМ как средство обучения». Студенты знакомятся с оборудованием КВТ, характеристиками используемых в них ЭВМ, существующими педагогическими программными средствами (ППС) и их применением в учебном процессе.

Не удовлетворенная пока что потребностью в ППС по различным школьным предметам, проведение факультативов и кружковой работы по программированию, большая воспитательная роль выполнения работ с применением ЭВМ требуют от будущих учителей информатики и вычислительной техники также знаний и умений в проектировании школьных ППС. Имеется в виду необходимость разработки некоторых ППС самими учителями (вместе со школьниками старших классов) хотя бы для нужд собственной школы. Лучшие ППС могут быть представлены в организации, занятые созданием фонда школьных ППС.

Чтобы решить поставленную задачу, в программу указанного курса в ЛГПИ им. А. И. Герцена включено изучение методов проектирования ППС с проведением необходимых практических и лабораторных работ. Изучаются методы диалогового проектирования автоматизированных учебных курсов в системе АСТРА-МИКРО, создания ППС по типу экспертных систем, метод автоматизированного проектирования ППС по структурированным образцам на языке MSX-Бейсик.

К сожалению, первые два метода на данной

стадии их развития не позволяют реализовать на школьных ЭВМ возможности машинной графики для повышения эффективности обучения за счет демонстрационных средств и игрового компонента. С помощью третьего метода эти возможности реализуемы.

Научиться составлять новые программы легче всего по готовым образцам, используя прототипы и накопленный опыт. По этому пути прошли многие программисты. При проектировании ППС по структурированным образцам исполнитель прежде всего знакомится с созданной библиотекой специально подготовленных, хорошо структурированных программ нужного ему типа. Если структура программы, выбранной в качестве прототипа, подходит, то исполнитель просто меняет ее содержательную часть. Например, известная обучающая программа по обществоведению «ПРАВО» легко переделывается в программу по географии или биологии, если разработчика удовлетворяет принятая в ней структура диалога с использованием словарика понятий, одними и теми же контрольными вопросами. Программа для тренировки в переводе английских слов на русский язык легко может быть переделана на перевод с французского или немецкого языка и т. д.

Выполнение простого задания, связанного с переделкой готовой программы, для непрофессиональных программистов является хорошим стимулом для дальнейшего совершенствования в области проектирования ППС и выполнения более сложных заданий.

Структурированная программа состоит из функциональных модулей, которые размещаются начиная с определенных номеров строк и имеют зарезервированные места для расширений. В ППС имеются обычно модуль с заставкой, модуль генерации заданий, ввода

и анализа ответов, блоков данных, игрового компонента и др. Связи между модулями минимизируются и фиксируются.

Если разработчика новой программы что-либо не устраивает в прототипе, он может произвести замену модулей, сам создать необходимый ему модуль или передать эту работу профессиональному программисту. Например, модуль ввода и анализа ответов с выбором из меню может быть достаточно просто заменен модулем со свободно конструируемым ответом и проверкой правильности ответа по ключевым словам. Часто появляется желание изменить игровой компонент в тренирующей программе. Для этого можно выбрать новый компонент из имеющейся библиотеки. Можно также создать новый игровой модуль. Для человека, знакомого со средствами машинной графики, решение этой частной задачи никаких трудностей не представляет.

Конечно, проектирование новых модулей подчинено определенным требованиям: зарезервированы имена переменных, фиксированы строки начала подпрограмм. Но возможен и другой подход, когда разработчик нового модуля просто будет следовать тому образцу, по которому был разработан прототип. Это особенно подходит для создателей новых игровых модулей. Например, игровой компонент с гонкой автомобилей при тех же размерах выделенного окна на экране дисплея и тех же именах переменных легко заменяется на парусную регату или переход верблюдов через пустыню и т. д.

Составление библиотек готовых структурированных программ и модулей — веление времени, первый шаг на пути создания систем автоматизированного проектирования ППС (САПР ППС). К этому необходимо приступить для повышения производительности труда в области проектирования ППС. Первый такой шаг в ЛГПИ им. А. И. Герцена уже сделан.

При создании САПР ППС ставилась задача: используя положительный опыт, накопленный в системах АОС ВУЗ, АСТРА-МИКРО и др., обеспечить дальнейшее развитие методов проектирования обучающих программ за счет возможностей машинной графики, разделения и распараллеливания работ по созданию программных функциональных модулей и проектированию программы в целом.

В состав оборудования САПР ППС в ЛГПИ им. А. И. Герцена входит обычное рабочее место учителя в кабинете вычислительной техники. Оно состоит из школьной ПЭВМ с монитором, дисководом и принтером. В качестве школьной ПЭВМ используется «Ямаха» с языком программирования MSX-Бейсик. САПР создается как система открытого

типа. Преподаватели и методисты при работе с САПР знакомятся с наличными структурами обучающих программ, отбирают необходимые им заготовки, заполняют их своим предметным материалом, включают при необходимости игровой компонент и с помощью команды MERGE производят «сборку» всей программы. Независимо от этого профессиональные программисты разрабатывают новые структурированные модули и пополняют ими библиотеку заготовок. Студенты и школьники создают игровые компоненты и другие несложные модули. К такого рода деятельности каждый из участников хорошо подготовлен, это дает значительное повышение производительности коллективного труда.

Системы автоматизированного проектирования — это развивающиеся системы. Они как бы аккумулируют в себе тот опыт, который накапливается со временем в результате труда многих разработчиков, и делают его всеобщим достоянием.

Созданная САПР является пока только началом работ в данном направлении. Невелик еще набор структурированных образцов. В них имеются модули с постоянной выборкой одних и тех же вопросов, с выборкой по случайному закону, с несколькими альтернативными ответами, с контролем правильности ответа по ключевым словам.

Адаптация по сложности изучаемого материала и тренировочных заданий осуществляется либо изменением выдаваемых обучаемому данных, либо изменением времени, отпускаемого на выполнение заданий. Предусмотрены средства адаптации и в контролируемых программах. Они позволяют получить оценку знаний и навыков с заданной точностью и достоверностью при минимальном для данного обучаемого количестве контрольных вопросов.

Игровые компоненты для тренирующих программ разработаны в расчете на одного или двоих обучаемых, работающих за одним дисплеем по очереди. В играх представлены два участника или два движущихся объекта, между которыми идет соревнование. Действия участников игры отображаются на экране и соответствуют успехам обучаемых. Победителем оказывается тот, у кого больше правильных ответов.

Если при тренировке за дисплеем работает один человек, то действующие персонажи перераспределяются. Один из них продвигается к цели на один шаг при правильных ответах, а другой при допущенных ошибках. По желанию разработчика окончание тренировки может быть отмечено исполнением музыкальной пьески.

Работа с САПР в учебном процессе началась в ЛГПИ им. А. И. Герцена в 1987/88 учебном году на IV курсе математического



факультета. После изучения языка программирования MSX-Бейсик со средствами машинной графики и программирования музыки студентам были выданы задания на разработку игрового компонента для тренирующих программ по предложенному образцу, но с новым сюжетом. На разработку сценария (сюжета), программирование, отладку, распечатку программы и составление отчета было отведено 8 ч учебного времени. Работа вызвала большой интерес в связи с ее творческим характером и производственной пользой.

Несмотря на довольно жесткие ограничения по характеру сюжета, размеру окна на экране, приемам программирования, разнообразию новых сюжетов, предложенных студентами, было поразительным: подъем альпинистов на Эльбрус, строительство дачных домиков, прогулка утят к озеру, парусная регата, выступления фокусников на арене, полет голубей мира и т. п. Практически две учебные группы в ходе обычного обучения без какой-либо оплаты и решения сложных организационных вопросов создали целую библиотеку игровых компонентов, внесли свой вклад в развитие САПР и создание новых ППС. Все студенты успешно справились с учебно-производственным заданием.

Для иллюстрации объема и трудоемкости учебного задания, связанного с разработкой игрового компонента, приводим распечатку программы «УТЯТА», разработанной студентками Власовой и Румянцевой.

```

1 'Игровой модуль "УТЯТА.ИГР". два утенка
2 'переходят дорогу, чтобы искупаться в озере
3 'Авторы: Власова, Румянцева - студентки
4 '4 курса ЛПИИ им. А.И.Герцена
10 SCREEN 2,2:COLOR 15,1,1:CLEAR 2000:CLS
20 OPEN "grp":"FOROUTPUT AS#:DEFSTR A,R
70 GOSUB 3000 'РИСОВАНИЕ ДЕКОРАТИВНОГО ФОНА
80 GOSUB 4000 'СОЗДАНИЕ ДВИЖУЩИХСЯ ОБЪЕКТОВ
130 PRESET (55,105):PRINT#1,"ПРИ НАЖАТИИ КЛАВИШ:"
140 LINE (8,116)-(22,132),15,B:PRESET (13,122)
145 PRINT#1,"1 - ИЗМЕНЯЕТСЯ ПЕРВЫЙ ОБЪЕКТ"

```



```

150 LINE (8,136)-(22,152),15,B:PRESET (13,142)
155 PRINT#1,"2 - ИЗМЕНЯЕТСЯ ВТОРОЙ ОБЪЕКТ"
160 LINE (8,152)-(38,172),15,B:PRESET (13,162)
165 PRINT#1,"TAB - ВЫХОД В САПР"
170 A$=INPUT$(1):B=ASC(A$)
180 IF B=49 OR B=33 THEN NO=1:GOSUB 5000:
GOTO 170
190 IF B=50 OR B=34 THEN NO=2:GOSUB 5000:
GOTO 170

```

```

200 IF B=9 GOTO 12000
210 GOTO 170

```

```

3000 'вид декоративного фона
3010 LINE (0,0)-(255,30),5,BF
3020 CIRCLE (25,10),5,10
3030 PAINT (25,10),10
3040 LINE (0,31)-(22,96),2,BF
3050 LINE (23,31)-(255,96),1,BF
3060 LINE (232,31)-(255,40),2,BF
3070 LINE (232,86)-(255,96),2,BF
3080 CIRCLE (255,63),23,7
3090 PAINT (255,63),7
3100 FOR I=0 TO 16
3101 LINE (22+I*12,41)-(28+I*12,86),15,BF
3102 NEXT I
3110 RETURN

```

```

4000 'создание движущегося объекта: утенок
4010 RESTORE 15000:F$="":FOR I=1 TO 32:READ F1
4015 F$=F$+CHR$(F1):NEXT I:SPRITE$(1)=F$
4020 PUTSPRITE1,(1,44),4,1
4030 PUTSPRITE2,(1,67),8,1
4040 RETURN

```

```

5000 'поведение объектов
5010 IF NO=1 THEN H=44 ELSE H=67
5020 FOR I=1 TO 19:G(N0)=G(N0)+1
5030 PUTSPRITE NO,(G(N0),H),NO*4,1
5040 IF G(N0)>238 THEN 6000
5050 NEXT I
5060 RETURN

```

```

6000 'озеро
6010 LINE (0,100)-(255,191),1,BF
7000 'конец тренировки
7010 PRESET (80,139):COLOR 8:

```

```

PRINT#1,"К О Н Е Ц !"
7020 FOR I=0 TO 700:NEXT
7030 LINE (0,99)-(255,191),1,BF
7040 PRESET (10,122):

```

```

PRINT#1,"ПРИ НАЖАТИИ КЛАВИШ:"

```

```

7050 PRESET (3,142)
7055 PRINT#1,CHR$(34)+"ПРОВОЕЛ"+
CHR$(34)+" -ПОВТОРЕНИЕ"

```

```

7060 PRESET (3,160)
7065 PRINT#1,CHR$(34)+"TAB"+CHR$(34)+
"ВЫХОД В САПР"

```

```

7070 A$=INPUT$(1):B=ASC(A$)
7080 IF B=32 THEN 10 'НА ПОВТОРЕНИЕ
7090 IF B=9 THEN 12000 ELSE 7070:'ВЫХОД В САПР
12000 'выход в САПР
12010 LINE (90,178)-(160,190),4,BF:
PRESET (105,180):COLORB

```

```

12020 PRINT#1,"ИДИТЕ":LOAD"САПР",R
14999 'блок данных для спрайтов
15000 DATA 0,0,1,1,0,128,128,191,255,127,127,
63,3,3,3,0
15010 DATA 224,240,252,248,240,96,96,224,240,
240,240,224
15020 DATA 224,0,128,192,128

```

Для размещения игрового компонента в верхней части экрана выделено 96 строк. Часть программы, связанная с изображением декоративного фона (переход через дорожку,

озеро), должна размещаться в строках 3000—3999. Строки 4000—4999 отводятся для создания движущихся объектов. Строки 5000—5999 занимают команды перемещения персонажей. Блоки данных размещаются начиная со строки 15 000. Чтобы игровой компонент можно было посмотреть в работе отдельно, в модуль включены строки 10—110

**В. ЕФИМОВ**

## Компьютер в педвузе

К написанию этой статьи автора побудило совещание-семинар проректоров по учебной работе и заведующих кафедрами информатики педагогических институтов РСФСР «Информатика и вычислительная техника в учебном процессе и управлении», которое состоялось в феврале 1989 г. в г. Омске. Проблемы, обсуждавшиеся на нем, профессионально близки автору, ученому секретарю секции «Подготовка и повышение квалификации педагогических кадров» Общесоюзного научно-методического совета новых информационных технологий (НМС НИТ).

Четвертый год педвузы страны ведут подготовку учителей информатики и вычислительной техники на базе специальностей «Математика», «Физика», «Общетеchnические дисциплины и труд». В 1988 г. около полутора тысяч выпускников получили дипломы, в которых графа «Квалификация специалиста» стала выглядеть так: «Учитель математики, информатики и вычислительной техники», «Учитель физики, информатики и вычислительной техники». В 1989 и последующих годах выпуск учителей нового профиля значительно увеличится. Предполагается, что в 1990 г. такую квалификацию получит около 10 тыс. молодых педагогов, прошедших полный курс компьютерной подготовки, способных вести преподавание в школе предмета «Основы информатики и вычислительной техники».

Как же у нас обстоят дела в этом новом для педагогических институтов направлении? Как используется вычислительная техника в учебно-воспитательном процессе? Эти вопросы обсуждались в г. Омске.

Совещание-семинар приняло рекомендации по обсуждаемому проблемам, которые в ближайшее время будут опубликованы для системы педвузов. Было отмечено, что основные направления, содержание и структура компьютерной подготовки педагогов зависят

и 6000—7050. При компоновке ППС команды этих строк не используются.

Работы по развитию и дальнейшему совершенствованию САПР ППС продолжаются. Преподаватели, программисты и студенты вместе решают важную задачу постепенного перехода на новые компьютерные методы обучения.

от того, как будет развиваться школьная информатика. Будет ли информатика (в IX—X кл., как это есть сейчас, или в более младших классах) предметом специального изучения или будет интегрироваться со школьными предметами?

Педвузы должны быть готовы к любому повороту событий. Впрочем, гибкость и разнообразие форм и содержания обучения будущих педагогов должны определяться не только этими факторами, но и потребностью в учителях определенного профиля для каждого конкретного региона.

Не вызвала сомнений у участников совещания-семинара необходимость формирования информационной культуры, умений и навыков использовать компьютер как мощное и эффективное средство обучения независимо от того, будет ли информатика отдельным предметом или нет. Поэтому требуются переосмысление содержания профессионально-предметной подготовки учителей всех специальностей, постановка их обучения на рельсы информационных технологий.

Объективно такая постановка вопроса еще 3—4 года назад была невозможна. Сейчас педвузы с учетом имеющегося опыта имеют для этого хоть и очень скромные, но реальные условия.

Инициатором активного внедрения в учебный процесс педвуза компьютерно-ориентированной подготовки студентов всех профилей был Омский госпединститут им. А. М. Горького.

Энтузиасты-преподаватели под руководством проректора института профессора М. Лапчика провели эксперимент по внедрению в учебные планы и программы специальностей нефизико-математического профиля блока новых учебных дисциплин, по замыслу обеспечивающих компьютерную грамотность студента, подготовку будущего педагога к профессиональной деятельности. Непрерывная и комплексная компьютерная подготовка студентов основывается также на компьютерной ориентации других компонентов педагогической подготовки учителя: педа-

гойки, психологии, частных методик, базовых дисциплин предметной специализации.

Опыт Омского пединститута нашел отражение в рекомендациях совещания-семинара по пересмотру учебных планов нефизико-математических специальностей педвузов, где должна быть предусмотрена непрерывная компьютерная подготовка студентов за счет введения таких дисциплин, как общеобразовательный курс информатики, основы информационной технологии обучения и др.

Характерной особенностью компьютерной подготовки учителей-предметников нефизико-математического профиля является пользовательский уровень применения ЭВМ в учебном процессе. Здесь не ставится задача готовить из будущего учителя химии, истории, иностранного языка и т. д. программиста. Главное — умения и навыки работы в условиях, когда в традиционную схему «учитель-книга-ученик» внедряется компьютер (как средство передачи информации и как средство изучения конкретного школьного предмета) с развитой системой педагогических программных средств (ППС) и их соответствующего методического обеспечения. Однако пользовательский уровень не ограничивает участие студента или учителя в разработке ППС, наоборот, программист-разработчик найдет помощь у учителя-предметника в методической аранжировке разрабатываемой программы.

Другое направление компьютерной подготовки будущего учителя, как отмечено выше, связано с обучением студентов физико-математического профиля информатике для преподавания школьного курса. На совещании-семинаре было подчеркнуто, что, учитывая потребности общеобразовательной школы в педагогических кадрах, способных на должном уровне преподавать эту дисциплину, целесообразно сохранить в основном сложившуюся с 1985 г. систему подготовки учителей математики и физики по дополнительной квалификации учителя информатики и вычислительной техники. Педвузам, однако, предоставлена и самостоятельность в определении структуры и содержания компьютерной подготовки таких педагогов.

Здесь, конечно, нельзя не учитывать позитивный опыт, накопленный в ряде педвузов: Свердловском, Омском, Ленинградском, Московском областном, Казанском. Он показывает необходимость внесения корректив в структуру непрерывной и комплексной подготовки учителя информатики, усиления методологического и прикладного характера блока компьютерных дисциплин, внедрения новых информационных технологий в учебно-воспитательный процесс. Что особенно важно для профессионализации подготовки

учителя информатики на базе физико-математических специальностей, так это приведение содержания базовых дисциплин, определяющих профессионально-предметную подготовку, в соответствие с возможностями использования в их преподавании вычислительной техники. С этой целью необходимо в ближайшее время пересмотреть соответствующие учебные программы, расширить и углубить работу по созданию учебно-методических материалов, программных средств, составляющих основу информационной технологии обучения.

Таким образом, складывающаяся практика подготовки педагогов нового типа внешне выглядит обнадеживающе.

Накоплен определенный опыт такой работы, есть понимание сделанного и перспектив совершенствования, значительно увеличился парк вычислительной техники педвузов, нарабатываются программный продукт, учебно-методические материалы. Но проблем и нерешенных вопросов значительно больше, чем решенных.

Ограниченность сегодняшнего понимания проблемы и соответствующих шагов по внедрению в учебный процесс информационных технологий прежде всего связана с общим крайне низким состоянием информатизации других сфер общественной и экономической жизни в стране. По выражению академика А. П. Ершова, мы находимся только в начальной стадии создания «инфосферы», общества составной частью которой является инфоструктура образования.

В системе подготовки педагогических кадров нового поколения сложности прежде всего связаны с материально-техническим обеспечением. Безусловно хорошо, что все пединституты и ИУУ оснащаются однотипной техникой, правда, японского производства. Можно решать задачи компьютеризации с единых позиций. И все же есть «но»! Техники этой мало. Что такое 1—2 учебных комплекта в варианте 9+1 (9 ученических мест и один головной компьютер) с одним дисководом и одним принтером, — хорошо знают в педвузах. Этого с трудом хватает только физико-математическому факультету. Фактически отсутствует периферийное оборудование. Казалось бы, действительно хорошо, что у всех пединститутов есть «Ямахи», а между тем это вариант без перспективы (на 5—6 лет). Это стало очевидным после трех лет эксплуатации первой партии поставки 1985 г. не только потому, что выходят из строя дисководы и разваливаются дискеты. Дело в принципе: педвузы и ИУУ со своими компьютерами «Ямаха» уже сейчас оказались в изоляции. Образовалась замкнутая подсистема внутри системы народного образования.

Кроме того, возможности этой техники не позволяют создать удовлетворительную учебную базу данных и базу знаний, без чего вообще нельзя вести речь об информационной технологии обучения.

В школе и непедagogических высших и средних учебных заведениях «Ямах» практически нет. А учителей информатики, как впрочем и других профилей, учим на ней. Сами себе создаем дополнительные трудности!

Все же и в рамках этих ограниченных возможностей можно и необходимо решать как достаточно общие, так и частные вопросы. Требуется, и уже стала крайне необходимой, координация деятельности педвузов по линии разработки, экспертизы, тиражирования и распространения программных средств и сопутствующих материалов, ориентированных на подготовку учителя. Сейчас каждый институт работает изолированно, обмен разработками происходит на личных контактах, из любезности и доброжелательства. Неизбежно дублирование, а значит, лишние затраты.

Есть попытки решения этого важного вопроса в рамках регионального учебно-методического центра по компьютеризации педагогического образования на базе какого-либо педвуза, но мешают трудности финансового характера, так как требуются дополнительные штаты сотрудников, финансирование разработок, солидный парк вычислительной техники и другое оборудование.

Следует также решать вопросы координации и на общесоюзном уровне.

Незамедлительных совместных действий вузов и органов управления требуют и другие важные вопросы: организация при ведущих пединститутах и университетах дополнительных ФПК, стажировки, аспирантуры и докторантуры по проблемам информатики в школе и вузе и по методике ее преподавания, методике обучения различным дисциплинам с применением ЭВМ; расширение практики направления студентов старших курсов, склонных к научно-исследовательской деятельности, в ведущие педвузы и университеты для завершения образования с последующим распределением на преподавательскую работу в рекомендовавший вуз; организация различных семинаров, школ для учителей и преподавателей педвузов по проблемам информатики.

Справедливости ради следует признать, что не только внешне-объективные причины тор-

мозят перестройку работы педагогических институтов в области новых информационных технологий обучения. Медленно, с опаской принимают новые идеи преподаватели. Трудно ломать привычные, установившиеся традиции в формах и методах обучения, преодолевать психологический барьер, возникающий из-за отсутствия личного опыта общения с компьютерами. Но самое поразительное в том, что довольно консервативными в освоении компьютерной технологии оказались преподаватели-математики. Физики, химики, биологи, филологи и многие другие не только быстрее оценили новые возможности в использовании готовых программных средств, но и активно включаются в их разработку, не будучи специалистами в области программирования. А математика что же? Разве мало возможностей имеет ЭВМ в преподавании математического анализа, геометрии, алгебры? Нельзя же ограничиваться «компьютеризированным» курсом «Численные методы». Надо все же отдать должное математикам за их активную работу по обеспечению компьютерных дисциплин. Здесь, безусловно, сделано много. Но это касается обеспечения внутренней потребности базового курса информатики, а надо идти дальше. Хотя математикам, конечно, сложнее. Учитель литературы — потребитель информационной технологии, а учитель математики и одновременно информатики — и потребитель, и изготовитель технологии. Пока это так. Как будет?

Еще один наболевший вопрос. Приходится слышать частые ссылки на острый недостаток в пединститутах вычислительной техники.

Но везде ли она используется с максимумом отдачи? Нередко можно видеть такую картину: закрытые сутками на шифрованные замки двери лабораторий и учебных кабинетов.

Решение острых вопросов зависит от всех нас: преподавателя, студента, руководителя, органов управления, а также от сбалансированности и скоординированности усилий региональных и центральных советов по компьютеризации образования. Искать нужно вместе, сообща, не кивая друг на друга, на высокое начальство или на нерадивость изготовителей техники. Школе, даже при нынешней ситуации с оснащением вычислительной техникой, требуется грамотный педагог, владеющий современной технологией обучения.

**А. РЕВУНОВ**

Рига

## Школьники выбирают дорогу

Для развития творческих способностей по информатике учащиеся СШ № 74 Риги объединены в научное общество (НОУ), которое состоит из двух секций — программирования и робототехники.

Членом научного общества может стать ученик, увлеченный вычислительной техникой или робототехникой, самостоятельно собирающий или изучающий материал по компьютерам, подготовивший обобщенный доклад, программу, разработку или конструкцию по вычислительной технике.

Научное общество возглавляет совет, в состав которого входят: учитель информатики, шефы-консультанты, ученые вузов, учащиеся, родители — специалисты по вычислительной технике. Совет занимается координацией и руководством работы секций.

Научное общество ведет большую работу по пропаганде знаний о вычислительной технике среди учащихся. Наиболее интересны такие формы и методы работы, как экскурсии, кинофестивали, конкурсы проектов «ЭВМ будущего», которые развивают творческую активность и самостоятельность школьников. Все работы, представленные на конкурс, вывешиваются на специальном стенде для ознакомления. В конце четверти проекты оцениваются специальным жюри из старшеклассников.

Школьники предлагают различные проекты: «Электронная машина времени», «ЭВМ — справедливый экзаменатор», «ЭВМ — управляющая система роботов для уборки квартиры», «ЭВМ — справочник-путеводитель по городу». Бывают шуточные проекты: ЭВМ — универсальный «подсказчик» на уроках, игровой автомат и др.

Старшеклассники из НОУ часто проводят интересные мероприятия в подшефных

классах. Их подготовка и проведение приносят большую пользу не только младшим учащимся, но и самим шефам. Ребятам нравятся игры с микрокалькуляторами «Угадай число», «Определи расстояние между городами», «Смешной марафон». С ЭВМ можно поиграть в шахматы, шашки, кубик Рубика, лабиринт.

Защита докладов «Компьютер в моей будущей профессии» перед товарищами также вызывает интерес школьников.

Конференция «1000 профессий ЭВМ» — одно из главных ежегодных событий в школе. Задачей участников является обсуждение возможностей использования ЭВМ в промышленности, медицине, в прогнозировании погоды и т. д. В рамках конференции проводятся конкурсы плакатов и стенгазет об ЭВМ. Популярен конкурс юмористических рисунков.

На конференции демонстрируются и обсуждаются программы по физике, информатике, математике и другим предметам.

В обсуждении принимают участие все ребята из НОУ. Лучшие работы с удачными техническими решениями выставляются в кабинете информатики на постоянной выставке, выдвигаются на городскую или республиканскую выставку технического творчества, а программы используются в школе и сдаются в Республиканский фонд алгоритмов и программ.

Членами НОУ, в частности, разработана программа профдиагностики учащихся. Опрашиваемые отвечают на вопросы анкеты. Сначала определяется увлеченность предметом, затем психологические особенности характера и состояние здоровья. Последний этап диалога связан с определением склонности к одному из пяти видов деятельности:

человек — человек, человек — техника, человек — природа, человек — художественный образ, человек — знак. Затем предлагаются профессии в соответствии с видом деятельности и необходимые качества, которыми надо обладать.

Далее с помощью принтера печатается ориентационный паспорт, в котором предусмотрена такая информация: профессиональные интересы, педагогические и психологические склонности, физическое раз-

витие и состояние здоровья ученика.

За последние годы члены НОУ трижды представляли свои работы на Всесоюзной неделе технического творчества школьников. Были участниками ВДНХ. Награждались медалями и дипломами.

Особенно важно, что почти все выпускники НОУ выбирают профессии, связанные с вычислительной техникой и теми отраслями знаний, которыми они занимались в секциях.

## Олимпиады по информатике

### II Всесоюзная олимпиада школьников по информатике

В апреле 1989 г. в Минске прошел заключительный этап II Всесоюзной олимпиады по информатике. В нем приняли участие 3 учащиеся VIII класса и 79 учащихся IX—X классов, составлявших 15 команд от союзных республик, команды Москвы, Ленинграда, Минска и Министерства путей сообщения. Оргкомитет заключительного этапа возглавлял министр народного образования БССР М. И. Демчук, председателем жюри был академик АН БССР В. А. Лабунов.

Соревнование проводилось в два тура — теоретический и практический (по четыре астрономических часа); задачи были одинаковыми для всех участников, независимо от возраста. На теоретическом туре давалось три задания, на практическом — два; можно было использовать ПЭВМ ЕС-1840 или «Ямаха» и любые имеющиеся языки. Перед практическим туром участники имели возможность поработать на ПЭВМ для приобретения практических навыков. Решения задач требовалось снабдить пояснениями: коротко и ясно изложить основную идею алгоритма, описать форму представления данных, используемый алгоритм, особенности его представления в виде программы.

При проверке работ учитывалась правильность решения задачи и эффективность предложенных алгоритмов. Оценивались решения по 10-балльной системе. За наиболее интересные решения жюри добавляло к основной оценке 1—5 баллов.

Всего было присуждено 6 дипломов первой степени, 9 — второй и 10 — третьей; ряд участников были награждены специальными призами. Дипломы первой степени получили следующие участники (в скобках — набранные баллы, школа, город, учитель):

Илья Долголяцкий (49,5; № 239, Ленинград, Е. П. Смолянинова);

Лев Новик (38; № 239, Ленинград, Е. П. Смолянинова);

Вейко Саар (37,5; Нысская СШ, ЭССР, У. Пусс);

Илья Жильцов (36,5; № 37, Свердловск, О. А. Черноус);

Дмитрий Козлов (34,0; № 556, Ленинград, Е. И. Забоклицкий);

Георгий Чесаков (30,5; № 7, Москва, Е. В. Юрченко).

В неофициальном командном зачете первые места заняли команды Ленинграда (40), ЭССР (29), Москвы (24) (в скобках — средние баллы членов команды).

Приведем теперь задания олимпиады. В скобках после формулировок заданий указаны: число участников, получивших за задание более 9 баллов; число участников, получивших 0 баллов; фамилия автора задания.

#### Теоретический тур

Задание 1. По итогам работы составлены два списка по оценке трудового участия одних и тех же  $N$  сотрудников ( $1 \leq N \leq 6$ ): первый — по решению трудового коллектива, второй — по мнению начальника. Начисленная сотруднику премия определяется по формуле:

$$S(K) = K(N+1)^{M-1} + 2K + 3M,$$

где  $K$  — номер сотрудника в списке коллектива,  $M$  — его номер в списке начальника.

Зная лишь сумму премий всех сотрудников и первый из списков, составить алгоритм, позволяющий установить второй список (11; 2; Н. Н. Красовский).

Решение. Суммарная премия равна:

$$S = \sum_{k=1}^6 I_k(N+1)^{J_k-1} + 5 \frac{N(N+1)}{2},$$

где  $I_k$  — номер  $K$ -го человека в первом списке,  $J_k$  — номер  $K$ -го человека во втором списке. Тогда

$$\sum_{k=1}^6 I_k(N+1)^{j_k-1} = S - 5 \frac{N(N+1)}{2}.$$

Число в левой части равенства можно рассматривать как некоторое число в  $N+1$ -ричной системе счисления; тогда цифры  $I_k$  этого числа можно получить, последовательно выполняя над ними операции поиска остатка от деления на  $N+1$  и деления без остатка на  $N+1$ .

**Задание 2.** Структура простого предложения имеет вид:

(определение 1) подлежащее сказуемое (определение 2) (дополнение) (обстоятельство).

Члены в круглых скобках могут отсутствовать.

Сформулировать правила составления простого предложения и предложить алгоритм, генерирующий по ним все простые предложения из заданного вами словаря. Словарь состоит из четырех групп слов (в каждой — не менее двух слов): существительные, прилагательные, глаголы, наречия (6; 13; Ю. С. Завальский).

**Решение.** Необходимо описать некоторые правила согласования частей предложения и осуществить перебор различных вариантов слов из словаря с учетом структуры предложения.

**Задание 3.** На плоскости задано множество из  $N$  прямоугольников, стороны которых параллельны осям координат. Каждый прямоугольник задается координатами его левой нижней и правой верхней вершин. Составить алгоритм определения наибольшего натурального числа  $K$ , для которого существует точка плоскости, принадлежащая одновременно  $K$  прямоугольникам.

**Примечания.** Эффективным считается алгоритм, число действий в котором пропорционально  $N^3$ . Точка, лежащая на границе прямоугольника, принадлежит ему. Упрощенный вариант: прямоугольники заменить отрезками на оси  $Ox$  (8; 44; А. Шень).

**Решение.** Оно сводится к решению задачи на сетке, узлы которой образованы линиями, параллельными осям координат и проходящими через вершины прямоугольников. На первом этапе следует упорядочить «иксовые» координаты прямоугольников в порядке убывания левых координат, а если левые координаты равны, то в порядке убывания правых координат. При этом для каждой пары координат следует знать номер прямоугольника. Затем для каждой горизонтальной линии построенной сетки выполнить следующие действия:

1. Определить номера прямоугольников, пересекаемых линией.

2. Для каждого узла сетки, лежащей на прямой, определить число, скольким парам «иксовых» координат она принадлежит. При этом следует учитывать, является ли точка началом либо концом некоторой пары. Если она является началом пары, соответствующей прямоугольнику, пересекаемому линией, то мощность пересечения увеличивается на 1. Если она является концом пары, соответствующей прямоугольнику, пересекаемому линией, то на следующем узле сетки мощность пересечения уменьшается на 1.

### Практический тур

**Задание 1.** Написать программу определения количества билетов с  $2N$ -значными номерами, у которых сумма первых  $N$  десятичных цифр равна сумме последних  $N$  десятичных цифр;  $N$  — произвольное натуральное число. Программа должна вывести на экран последовательность искомых количеств для  $N=1, 2, 3, \dots$ . При оценке программы учитывается количество выведенных чисел (количество отработанных  $N$ ) (6; 13; В. М. Котов).

**Решение.** Для эффективной реализации алгоритма необходимо воспользоваться следующими соображениями.

1. Пусть  $S(I, J)$  — количество  $I$ -разрядных чисел, сумма цифр которых равна  $J$ . Тогда

$$P(I) = \sum_{J=0}^{9I} S(I, J) \cdot S(I, J),$$

где  $P(I)$  — количество  $2I$ -разрядных «счастливых» билетов.

$$2. S(I+1, J) = \sum_{K=0}^{\min(J, 9)} S(I, J-K).$$

$$3. S(I, J) = \begin{cases} 1, & 0 \leq J \leq 9 \\ 0, & J > 9. \end{cases}$$

4. Необходимо моделировать «длинную» арифметику.

**Задание 2.** На столе лежит игральный кубик гранью  $M(0)$  к нам, гранью  $L(0)$  вверх. Написать программу, определяющую последовательность «кантований» кубика («на нас», «от нас», «вправо», «влево»), после выполнения которых кубик окажется на прежнем месте, но гранью  $M(K)$  к нам, а гранью  $L(K)$  вверх.

**Примечания.** Под «кантованием» понимается перекачивание кубика через соприкасающееся со столом ребро без скольжения. Другие способы перемещения кубика запрещены. Нумерация граней кубика такова, что если его положить на грань с цифрой 5, то на боковых гранях будут цифры 1, 6, 4, 3 при обходе по часовой стрелке, а на верхней — цифра 2 (1; 30; В. В. Прохоров).

**Решение.** Его можно получить, вращая куб-

бик вокруг диагоналей (т. е. когда некоторая точка остается неподвижной), причем достаточно вращений вокруг двух диагоналей, которые определяются начальным и конечным состояниями.

Если за два вращения не удалось получить искомым результат, то задача неразрешима.

С. Бондаренко

## Международная олимпиада

Первая международная олимпиада школьников по информатике состоялась в болгарском городе Плевне 17—19 мая 1989 г. В олимпиаде приняли участие 16 команд из 13 стран. Каждая команда состояла из трех участников. Болгария была представлена двумя командами, СССР — тремя. Первая советская команда — сборная страны, образованная в результате отбора из участников летних сборов, проходивших в Химках 3—16 мая. В нее вошли Илья Долголяцкий (Ленинград), Лев Новик (Ленинград), Андриюс Чяпайтис (Каунас). Руководитель команды — доцент МИФИ В. М. Кирюхин. Вторая сборная представляла Переславль-Залесский; третья — наше посольство в Париже.

Научная комиссия, состоявшая из выдающихся болгарских ученых, получила к моменту открытия олимпиады шесть задач. Жюри формировалось по демократическому принципу: в него вошли члены комиссии и все руководители команд. Утром 17 мая, пока участники знакомились с компьютерами, на которых им предстояло состязаться, жюри рассмотрело все представленные задачи и выбрало одну из них. После осуществления перевода и подготовки текста для каждого участника началось само состязание. На работу было отведено четыре часа. В распоряжение каждого участника организаторы предоставили персональный компьютер «Плевнец» с языками программирования Бейсик, Паскаль, Си. Наибольшей популярностью пользовался Паскаль. От участников требовалось представить не только отлаженную программу, но и доказательство справедливости выбранного алгоритма. Максимальное количество баллов, которое мог получить участник, — 100. По окончании отведенного времени участники сдавали жюри свои программы на магнитных носителях и распечатки исходных текстов.

Из команд были составлены отдельные группы, к каждой из которых прикреплены несколько членов методической комиссии.

18 мая весь день продолжалась работа жюри по группам. 19 мая утром состоялось объединенное заседание жюри, на которое были предложены для рассмотрения оценки по каждой группе с кратким обоснованием.

В итоге было решено наградить дипломами первой степени участников, набравших 95—100 баллов (пять дипломов), дипломами второй степени — набравших 80—90 баллов (шесть дипломов), дипломами третьей степени — набравших 60—79 баллов (семь дипломов). Наша сборная получила два диплома: первой степени — Андриюс Чяпайтис (95), второй степени — Лев Новик (80).

Закрытие олимпиады состоялось 19 мая в городском Дворце культуры. Были вручены дипломы и призы победителям. Специальными призами были награждены двое участников: девушка из Греции как единственная представительница женского пола среди официальных участников; самый юный участник олимпиады — член нашей команды из Парижа, набравший 40 баллов.

Председатель жюри Петр Кендеров объявил, что в 1991 г. олимпиада пройдет в Греции и в 1992 г. — в ФРГ; место проведения олимпиады 1993 г. вакантно.

Э. Ратобыльская

## Задача Международной олимпиады по информатике

Дана последовательность из  $2N$  ячеек. Две соседние из них — пустые, а в остальных расположены  $N-1$  символов А и  $N-1$  символов В.

Пример для  $N=5$ :

А	В	В	А			А	В	А	В
---	---	---	---	--	--	---	---	---	---

*Правило перемещения.*

Содержимое любых двух соседних пустых ячеек можно, сохраняя их порядок, пересылать в пустые ячейки.

*Цель.*

Используя правило перемещения, достичь конфигурации, в которой все символы А расположены левее всех символов В. Местоположение пустых ячеек после перемещений не имеет значения.

*Задача.*

Составить программу, которая:

1) вводит с клавиатуры начальную конфигурацию в виде последовательности символов А, В и нулей для пустых ячеек, а также моделирует перемещения;

2) для заданной начальной конфигурации



определяет по меньшей мере один план перемещений, с помощью которого можно достичь цели, или сообщает, что такой план не существует. Вывод должен содержать начальную конфигурацию, промежуточные конфигурации после каждого шага, а также

заключительную конфигурацию;

3) находит некоторый план, достигающий цели за минимальное число шагов.

*Результаты.*

Представьте по крайней мере одно решение для примера, приведенного выше.

---

## **Вторая международная олимпиада школьников по информатике**

состоится в Минске (БССР) 15—21 июля 1990 года. Организационный комитет по проведению олимпиады будет признателен странам-участницам, которые любезно согласятся направить в адрес оргкомитета варианты задач, предлагавшихся на национальных состязаниях, а также свои предложения и пожелания по организации и регламенту олимпиады.

Дополнительную информацию можно получить по телефонам:

**в Минске — 20-85-44,  
в Москве — 237-86-24.**

**Адрес оргкомитета олимпиады:  
220010, БССР, Минск, ул. Советская, 9.  
Министерство народного образования БССР.**

## **SECOND INTERNATIONAL PUPILS' OLYMPIAD ON INFORMATICS**

will be held in Minsk, Byelorussian SSR, on July 15—21, 1990.

The Olympiad Organising Committee will appreciate countries-would-be-participants' sending to the Committee samples of problems offered for solving in national events of the kind, as well as proposals and suggestions on organisation and schedule procedures.

Further information can be obtained

by phone: **20-85-44, Minsk**

**237-86-24, Moscow**

by mail: Olympiad Organising Committee,  
BSSR Ministry of People's Education,  
9, Sovetskaya Str.,  
Minsk, 220010, USSR

---

Д. ВОЛКОВ

## Школа, программирование, познание

*«Нас не профессии надо учить, а способности самостоятельно мыслить».*

А. Михалков-Кончаловский

Обоснованность введения специального предмета ОИВТ в школьную программу вызвала сомнения отдельных специалистов еще до принятия этого решения [1], однако желание форсировать события и получить мгновенную отдачу оказалось сильнее требований здравого смысла. Всеобщая компьютеризация школьного образования, таким образом, отметит скоро свой пятилетний юбилей.

Утверждать, что только умение программировать способно расширить сферу использования компьютеров для целей образования, — это утверждать, что изучение автомобиля способствует облегчению выбора на карте оптимального маршрута движения (т. е. связь, безусловно, есть, но не настолько линейная, как принято считать). Можно быстро научиться составлять сложные программы и использовать все вычислительные мощности компьютера и в то же время плохо читать, испытывать затруднения при поиске своей страны на географической карте, беспомощно разводив руками перед простой, но поставленной в нестандартной форме задачей. «Специалист» зачастую уже не в состоянии критически оценить результаты, полученные машиной, не может вывести приближенные числовые оценки решения той или иной задачи. Наблюдается потеря «физической интуиции», вместо которой появилось умение «заставить» компьютер выдавать числа. Да и эффективность использования ЭВМ не гарантируется самим фактом ее присутствия в классе. Из истории развития промышленного производства известно: «форми-

рование подготовки производства без предварительной уверенности в конечном продукте явилось скрытой причиной многих и многих катастроф». Весьма опасны и чреватые последствиями «наивные» предположения, будто компьютер — это «палочка-выручалочка» [2].

Сторонники преподавания программирования приводят ряд доводов в защиту своей точки зрения. Приведем некоторые из них.

Невозможно без программирования хорошо освоить компьютер, научиться работать с ним, избавиться от страха перед машиной. Иногда напоминают о той радости и чувстве удовлетворения, которые испытывают дети, когда видят, что компьютер подчиняется их командам. Программирование способствует алгоритмизации мышления, систематизации и планированию действий. Наконец, изучая программирование, школьник готовит себя к будущей профессии.

Не безмолвствуют и противники массового преподавания программирования в школе. Они приводят примеры проявления признаков оглуления, вызванного увлечением программированием. Говорят о том, что возникает многочисленная группа «программистов-функционеров», людей, которые основное свое внимание направляют не на решение задач, а на реализацию различных программистских трюков. Они зачастую бьются уже не в состоянии «взлететь» над проблемой, оглянуться вокруг и, возможно, обнаружить, что в данном случае и не нужно писать программу, а лучше подойти к задаче иным способом.

Слышатся доводы против увлечения программированием и со стороны медиков. Нельзя забывать, что появление первых компьютеров в школе сопровождалось,

естественно, повышенным интересом к ним, ученики забрасывали занятия спортом, отказывались от прогулок и осаждали компьютерные классы.

Стремительность старения техники и появления новых технологий сводит на нет большинство навыков программирования, полученных учеником в школе. Преподаватель может считать свою работу успешной, если ученик только сформулирует свои требования, а реализацией пусть займутся программисты-профессионалы.

Ответ на вопрос, нужно ли программирование в школе, можно получить только на основе эксперимента, причем не всеобщего, как это оказалось с массовой компьютеризацией, а ограниченного и проведенного для разных контингентов учащихся. Последнее замечание направлено против широко распространенной ситуации, когда «мыслят локально, однако пытаются действовать глобально». Имеющийся двадцатилетний опыт использования ЭВМ в школах таких крупных научных центров, как Новосибирск либо Дубна, трудно признать полностью приемлемым для всеобщего распространения, учитывая высокий образовательный уровень населения и специальный отбор в учебные группы.

Итак, в средней школе, со средними учениками сформированы две группы. (Эксперимент проводился в СШ № 179 Москвы.) Первая группа («программисты») получала все или почти все для полного освоения искусства программирования. Перед преподавателем стояла задача обучить учащихся работе с одним или несколькими языками программирования. Разберем подробнее некоторые методические приемы по организации обучения в этой группе.

Основное и неотъемлемое требование — это использование в процессе обучения реальных задач, т. е., говоря словами Н. К. Крупской: «Не просто задачи — планы, а планы, связанные с реальной практической работой, которые проводит школа в жизнь. План не кооператива вообще, а данного школьного кооператива». Преодолевая недоверие и инертность, в роли заказчиков удалось заполнить школу, пединститут и программистский кооператив. Полученные задания сразу же были распределены среди учащихся. Далее изучение каждой новой темы велось на основе этих задач. Структурное программирование и проектирование не вообще программ, а именно структуризация задания, освоение управляющих конструкций языка, особенностей работы с процедурами и т. п. велось всегда с прицелом на основное полученное задание. Все это позволяет удержать темп обучения и

сохранять постоянную заинтересованность, причем учитель выступает в роли не надсмотрщика, а мудрого наставника, ориентирующего ученика в потоке информации. Основной побудительный мотив (желание решить поставленную задачу) стимулирует ученика на поиск новых путей достижения цели и, что очень примечательно, способствует формулированию требования к набору инструментов познания. Школьник либо сам ищет то, что ему требуется среди уже изученных методов, либо с нетерпением ждет объяснения новой темы. В результате к очередному уроку он приходит, имея определенные представления о том, что ему нужно получить от него.

В силу чисто технических трудностей в ходе проведения эксперимента не удалось полностью ответить на вопрос: с какого языка программирования следует начинать? Любой язык содержит в себе определенную и достаточно устойчивую схему. Все языки программирования, как бы различны они ни были, сводятся к единому базису — уровню машинных команд. Возможно, что изучение программирования следует начинать знакомством с Ассемблером. Процесс программирования на языке низкого уровня достаточно трудоемок и утомителен, поэтому у учеников закономерно появляется желание обобщить какие-то операции и, используя аппарат макросредств, построить более сложные и удобные языковые конструкции. Создается ситуация, когда школьник сам задает требования к инструменту и сам же его создает. Безусловно, ученику нет никакой необходимости на основе Ассемблера реализовывать полный комплект нужных языковых средств, но самое ценное в таком подходе, что пониманию своих потребностей поможет ученику выбрать наиболее подходящий для работы язык высокого уровня и свести до минимума время на его освоение, ведь, программируя на Ассемблере с использованием макросредств, он уже частично заново открыл для себя этот язык.

С самого начала обучения следует прививать школьникам основы общей культуры программирования. Сюда входят навыки детального обдумывания будущей программы, учет требований мобильности (использование, например, только стандартных конструкций языка), требования по наиболее наглядному оформлению программ. Нельзя забывать и о конкурентоспособности программных разработок, что предусматривает обязательное наличие документации, требования к надежности и эффективности. Любая программа должна содержать инструкции по использованию и иметь средства для выдачи исчерпывающей информации о всех

возможных последующих действиях пользователя. Если разрабатывается диалоговая программа, а для персональных компьютеров таких программ подавляющее большинство, необходимо очень тщательно прорабатывать сценарий диалога и язык общения машины с оператором. Существует ряд хорошо известных методов проектирования диалога (например, [4]), но не следует забывать, что программу разрабатывает все-таки школьник, и любые рекомендации по ее созданию должны выгладеть со стороны преподавателя именно как рекомендации, а не как приказы. Пусть ученик сам ищет наиболее удобные, с его точки зрения, методы ведения диалога, тем более что обычно будущие пользователи программы — это те же школьники. Данное замечание справедливо при изучении любой темы. Если ученик настаивает на своем подходе, то не нужно пытаться «ломать» его. Лучший вариант решения выявится при критическом сравнении реализаций одного и того же задания, предложенного несколькими учениками. Может быть, единственный случай, когда учителю требуется проявить твердость, — это пресечение попыток буквального копирования друг у друга, именно копирования, а не живого обмена идеями и мнениями.

При разработке программ, реализующих в диалоге с пользователем поставленные в задачах функции, стала очевидной необходимость выделения специальных программ-инструментариев — «кирпичиков», которые можно использовать при создании различных прикладных программ. В состав этого комплекса программ могут входить графические модули, системы управления диалогом, блоки работы с окнами на экране дисплея и ряд других. Отсюда неизбежно напрашивается вывод: необходимо выделить отдельную группу учащихся, имеющих ясную склонность к программированию, которые и будут заниматься реализацией инструментальных модулей (конечно, только в случае отсутствия уже готовых). Все это потребует от учеников более глубокого знакомства с вычислительной системой, тем более это полезно, так как часто свою будущую профессию эти ребята непосредственно связывают с ЭВМ.

До сих пор речь шла лишь о первой группе — группе программистов и о методических приемах организации ее работы. Эксперимент предусматривает создание второй группы — «непрограммисты».

Минимум программирования и максимум подготовки по методам формализации и анализа задач — вот основа организации работы со второй группой. Непосредственное

знакомство с компьютером для этой группы сводилось к получению информации по общей организации вычислительного процесса, сравнительному обзору языков программирования и изучению различных прикладных систем: клавиатурных тренажеров, игровых модулей, иллюстрирующих весь спектр возможностей компьютера, моделирующих и имитационных программ. На первых занятиях было трудно преодолеть чувство массового сознания: ведь рядом занимается группа «программистов», которая «взахлеб» пишет одну программу за другой. Для преодоления таких настроений достаточно было напомнить, как еще совсем недавно огромные надежды возлагались на АСУ, системы распознавания образов, синтезаторы речи, на применение ЭВМ для решения шахматных задач и как постепенно на смену шапкозакидательскому ажиотажу пришло понимание того, что процесс мышления, например, вряд ли поддается полной формализации и без человека компьютер всего лишь машина [5]. В конце концов школьники согласились с тем, что первичным является все-таки системный анализ задачи и поиск методов ее формализации.

Неотъемлемой частью учебного процесса для второй группы были регулярные обзоры периодических компьютерных изданий с целью выявления наиболее общих подходов к решению задач, накопления опыта всестороннего изучения проблем. Устраивались также совместные обсуждения работ, проделанных учениками первой группы. Основное направление таких диспутов — это оценка целесообразности проделанной работы, характеристика используемого интерфейса с пользователем, поиск дополнительных сфер применения созданных программ.

Как известно, большое влияние на результаты работы оказывают непосредственные предварительные контакты с пользователем и необходимым условием успеха является привлечение учеников к этой деятельности. Часто бывает невозможно получить от заказчика четкую постановку задачи, и требуется итеративный процесс уточнения, включающий в себя создание макетов демонстрационных программ. Роль посредников между группой «программистов» и пользователем пытались сыграть именно ученики второй группы. Небреженность познаниями из области программирования позволяет им, не вникая в тонкости программной реализации задачи, детально прорабатывать ее концептуальную модель. Свобода оперирования понятиями на уровне концепции и семантики способствовала развитию у школьников зачатков критического мировоззрения, свободного от догм и шабло-

нов. Кстати, именно среди учащихся второй группы наблюдалось некоторое повышение и уровня общей гражданской активности (выступления в печати по вопросам компьютеризации школы, участие в обсуждении концепции информатизации и т. п.).

Для учащихся второй группы компьютер не выступает в роли обычного технического средства обучения, как это происходит сейчас, а является инструментом изучения окружающей действительности. Как уже отмечалось, надо учиться пользоваться этим инструментом разумно, а не пытаться «забывать микроскопом гвозди». В каждом предмете заложено множество свойств, и в каждой конкретной обстановке требуется выделить наиболее существенные. Компьютер для школы — это не удобное средство отдыха, получения красивой распечатки ведомости выдачи зарплаты или расчета по готовой формуле физической задачи, а прежде всего это принципиально новый инструмент, позволяющий расширить интеллектуальные возможности будущего специалиста, представляющий собой уникальное средство для обучения не знанию совокупности фактов, а методу исследования.

С целью проверки истинности сделанных наблюдений в конце курса обучения среди участников обеих групп был проведен психологический опрос.

Исследование проводилось по специально разработанной методике. Школьникам были предложены два варианта логических задач (силлогизмов, из которых необходимо сделать верный вывод): первый содержит формализованные структуры (А, В и С), кванторы общности и существования, второй представляет собой аналогичный набор силлогизмов, но построенный на обыденных речевых суждениях (задачи и идея построения методики заимствованы у Л. Кэррола из книги «История с узелками»). Особенность задач из данных наборов состоит в том, что решающему (делающему заключение) предоставляется возможность выбрать свой вариант суждения о соответствии объемов понятий в посылках (А может быть больше В, меньше, равно; точно так же и для С). Подобная произвольность «априорной» оцен-

ки сохраняется и в остроумных текстовых силлогизмах Л. Кэррола. Соответственно в решении всего списка задач можно проследить, насколько дисциплинированно и последовательно сохраняют учащиеся ими же выбранные правила соотношения объемов понятий.

Сравнение первой («программисты») и второй («непрограммисты») групп учащихся показало, что задачи с формализованными понятиями решаются в целом примерно одинаково, в то время как текстовые силлогизмы решаются лучше учениками из второй группы. Больше половины из них придерживались при решении начальной стратегии. В первой группе только около 10 % школьников следовали одной стратегии решения, для остальных были характерны частая смена стратегий, размытое толкование условий задач и излишний формализм.

Безусловно, не все в процессе обучения происходило гладко: ненадежность техники и отсутствие условий для комплексного проведения исследования не всегда позволяли сохранить нужный темп работы и достигнуть всех поставленных целей. Кроме этого, данный эксперимент в силу недостаточного распространения и организационных трудностей не претендует на завершенность, однако заставляет более критически взглянуть на законодательно проводимый процесс массовой компьютеризации.

## Л и т е р а т у р а

1. *ЭВМ в школе — реальность наших дней: «Круглый стол»* журнала ЭКО // Наука, 1984, № 11.
2. *Машбиц Е. И.* Психолого-педагогические проблемы компьютеризации обучения. М.: Педагогика. 1988.
3. *Буланова Н. Л. и др.* Математическое моделирование и перспективы развития школьного образования. М.: ИПМ им. М. В. Келдыша АН СССР, 1987.
4. *Далингер В.* Диалоговые обучающие программы и требования к ним // Информатика и образование. 1988. № 6.
5. *Поспелов Д. А., Пушкин В. А.* Мышление и автоматы. М.: Сов. радио, 1977.

## Я выбрал систему профтехобразования

(интервью с директором Всесоюзного научно-исследовательского института профтехобразования В. В. Шапкиным)



— Виктор Васильевич, как Вы пришли в систему профтехобразования? Это был сознательный, случайный или вынужденный шаг?

— Начало моего трудового пути связано с общеобразовательной школой — я окончил Ленинградский педагогический институт им. Герцена. Вышел из него учителем физики, машиноведения и электротехники, был сразу же рекомендован в аспирантуру, по окончании ее оставлен в институте. Потом был период, связанный с партийной работой. Когда мне исполнилось 40 лет, я стал думать: функционер ли я, так сказать, на перспективу, или все-таки надо вернуться к тому, чем я занимался. Пожалуй, профтехобразование я выбрал потому, что думал: именно здесь я смогу реализовать свои возможности в научном плане. Кроме того, я всегда ощущал интерес к гуманитарным проблемам. Хотя диссертация была по экспериментальной физике, мне никогда не были чужды вопросы, связанные с социологией, экономикой, педагогикой.

— Какие проблемы в системе профтехобразования Вы видели, вступая на пост директора Всесоюзного научно-исследовательского института профтехобразования и была ли у Вас программа их решения?

— 1981 год был апогеем расцвета концепции профтехшколы как глубокой интеграции общего среднего образования с профессиональной подготовкой при мощном развитии материальной базы, обеспечивающей, как мне тогда казалось, хорошее профессиональ-

ное становление личности. Поэтому я шел не столько «исправлять», сколько работать в, на мой взгляд, хорошо отлаженной системе, используя свои возможности и знания.

Проблемы я почувствовал, став директором, и понял, что надо отработать свою концепцию, свой взгляд на решение достаточно глубоких противоречий, которые есть в современной профтехшколе.

Хотя сегодня есть мнение, что интеграция среднего и профессионального образования была ошибкой, я с этим не согласен категорически. Считаю, что эта концепция в основе своей была правильной. Я убежден в том, что интеграция старшей ступени общеобразовательной школы и профессиональной подготовки по профессиям, которые определяют научно-технический прогресс, является и наиболее целесообразной, и органичной. Во-первых, потому, что обучение на старшей ступени должно быть все-таки мотивированным, а профессия у человека — всегда самый мощный мотив. Во-вторых, профессиональная подготовка должна не просто ложиться на готовую общеобразовательную базу, а идти параллельно, дополняя ее. Но это верно не для всех профессий. Дело в том, что специфика большого производства — неоднородность, т. е. научно-технический прогресс внедряется не по всем отраслям одинаковыми темпами. Отсюда вывод: всегда есть профессии «уходящие», профессии массового типа на данный момент и перспективные. Для части (некоторых) профессий необходимо слияние

общего среднего и профессионального образования. Но если этот принцип механически перенести на профессии, не требующие по своей сути такой интеграции, то мотив не включится. В этом случае мы ребят, так сказать, силой будем заставлять изучать то, что на первом этапе их трудовой деятельности им не пригодится, а отсюда и все последствие.

Мы надеялись, что создадим свою учебно-производственную базу — это была концепция развития собственных учебно-производственных мастерских, но эта идея тоже оказалась нереальной. Она была оправдана, когда речь шла о достаточно простых орудиях труда, которыми действительно можно было обеспечить учебные заведения. Но когда речь идет о технологическом оборудовании, которое выпускается единицами, то требовать его на нашу базу без достаточной загрузки бесперспективно.

Я вижу выход, и получаю в этом поддержку, в развитии специального тренажеростроения, специальных систем, на которых можно было бы отрабатывать базовые профессиональные умения и навыки. Сегодня у нас эта отрасль не развита. Надо проектировать и сам тренажер, и технологическую оснастку для него — это дорого. Но если определить четкую политику на перспективу, то можно было бы все довести до экономической целесообразности и эффективности.

Далее. Я убежден, что нужны новые концептуальные подходы, касающиеся образования с точки зрения его экономики. У меня были встречи с представителями отраслевых министерств, где я пытался выяснить, почему же они не заинтересованы в нас. Оказывается, они считают, что мы по своей учебно-методической документации, по используемым методикам просто не успеваем за прогрессом. Поэтому наш заказчик, я имею в виду предприятия, все дальше отчуждается от процесса подготовки. На мой взгляд интеграция должна способствовать не замкнутости и автономности системы образования, а обратному — чтобы промышленность все время испытывала необходимость своего участия в учебном процессе.

— Насколько все же велико отставание?

— Например, в Минавиапроме — передовой отрасли в техническом оснащении — максимальный цикл отработки учебно-программной документации и соответствующих методик — год. Мы же называем срок в пять лет, а реально он еще больше. И этот разрыв, к сожалению, не имеет тенденции к сокращению.

Кроме того, сейчас вопросы планирования, содержания, выбора методов, форм и даже сроков обучения полностью решаем мы сами, находясь внутри системы профтехобразования и, шире, системы народного образования, а надо активнее подключать и коллективы учебных заведений, и трудовые коллективы предприятий заказчика, и отраслевые министерства. Надо знать прогноз развития отрасли, а отсюда и вытекающий социальный заказ. Функции и полномочия должны быть разложены на всех паритетным образом, по крайней мере, я так считаю.

И, наконец, о моем педагогическом взгляде на решаемые проблемы. Я шел в систему профтехобразования точно зная, что потенциальные возможности ребят, которые сюда приходят, немалые.

Есть люди с золотым умом, а есть люди с золотыми руками. В среднем образовании мы это не учитываем. К сожалению, наша школа ориентирована на запоминание, на структурирование знаний. Приходит ребенок-практик в мастерские — он, может быть, мыслит не так быстро и не так системно, но зато проблему он может решить практически — и в этом проявляет себя лучше других. Похвалили его — он и на уроке физики уже не будет последним. Мы ставим «четыре», нам говорят: вы натягиваете отметки. Да нет! Просто у него появилась вера в собственные возможности, а при этом раскрываются такие резервы! Я считаю, что мы мало говорим об этой социальной функции системы профтехобразования по отношению к подросткам, которые недополучили ни в семье, ни в школе, а она, пожалуй, самая важная.

Мои контакты да и мое непосредственное участие в психологических исследованиях показывают, что наши ребята имеют богатое воображение, что они лучше ориентируются в нестандартных психологических ситуациях, устойчивее и в моральном смысле. Все-таки вопросы подготовки к жизни в широком социальном плане профтехшкола неплохо решала, несмотря на все сегодняшние упреки в ее адрес. И может быть это более важно, чем строгая система знаний, логики и т. д.

— Я внимательно читала новую Концепцию профтехобразования, которая была представлена на Всесоюзный съезд учителей. И знаю, что она разработана под Вашим руководством и при Вашем непосредственном участии. Я обратила внимание, и это меня порадовало, что в ней большое место уделяется вопросам нравственного развития личности, гуманизации образования, тому, о чем Вы только что говорили. Но почему, тем не менее, в Концепции так мало сказано

о средствах новых информационных технологий? Какова их роль в реализации Концепции? Какие проблемы можно решить с их помощью?

— Да, Вы правильно обратили внимание. Все это неслучайно. Конечно, вопрос о новых информационных технологиях привлекательный, я бы сказал, модный, и тут мы могли бы выигрышно развернуться.

Мы считаем, что необходимо использовать средства новых информационных технологий для индивидуализации, дифференциации обучения, разработки корректирующих методик, учитывающая разную подготовленность учащихся и освоение ими материала и т. д. Но если мы не решим вопросы — и в Концепции мы специально сделали на это упор — демократизации, открытости нашей системы, ориентации ее на человека, все остальное может опять оказаться тем инструментарием, который не ляжет на благодатную почву. Поэтому компьютеры, на мой взгляд, будут играть ведущую роль на следующем этапе, когда мы эти задачи решим. Это во-первых.

Во-вторых, пока об использовании информационных технологий в сфере образования говорят все-таки с точки зрения когнитивного знания. Я имею в виду компьютер как средство хорошей структуризации знаний, их поиска, представления в удобной экранной форме и отсюда — формирования у учащегося в основном когнитивного знания. Такое знание нужно как ориентировочная основа действий, но у профтехшколы есть и своя специфика. Прежде всего, мы должны использовать компьютер для формирования конкретных профессиональных действий, операций, умений, навыков, и пока о компьютерной технологии в этом аспекте говорится совсем мало. В этой связи компьютер может рассматриваться как средство моделирования и понимания производственных процессов, для отработки умения принимать решения в нестандартных ситуациях. Я все время говорю о необходимости подключения к компьютеру моделей и макетов реальных пультов управления, с которыми учащиеся будут встречаться в процессе производственной практики. Здесь компьютер был бы незаменим. Но сегодня к такому использованию компьютера мы не готовы. Поэтому я этот сюжет откладываю на перспективу, хотя на последних наших совещаниях и семинарах разговоры об этом идут.

— Какие отделы института занимаются проблемами компьютеризации профтехобразования? Расскажите, пожалуйста, в чем суть основных направлений их исследований.

— Компьютеризацией системы профес-

сионально-технического образования наш институт занимается сравнительно недавно, около четырех лет. Задача для нас была поставлена что называется «на марше». Что удалось сделать? Я думаю, пока немного, но тем не менее мы выбрали для себя специфический путь: тренажеры, развивающие специальные психофизиологические, психофизические, специальные прикладные профессиональные навыки. Я думаю, что нам этот путь не перекроет никто — ни общеобразовательная школа, ни техникумы (у них своя специфика), ни вузы.

У нас есть специальная лаборатория тренажеров на базе ЭВМ и микропроцессорной техники, которая сегодня, как нам кажется, определила стратегию разработки тренажеров. С привлечением сторонних специалистов в лабораторию уже разработаны достаточно сложные и, мы надеемся, эффективные тренажеры по подготовке операторов сталеплавильного производства, химических производств, операторов станков с ЧПУ на базе ВТ отечественного производства.

Другая лаборатория института — лаборатория педагогических программных средств. Ее задача — найти формы, пути и возможности для разработки инструментальных средств. При этом мы ориентируемся на использование сервисных систем. Например, АСТРА/МИКРО — система, которая адаптирована и выполнена по нашему заказу МИЭТом. На ее основе уже сегодня получен достаточно широкий спектр прикладных программ. Я думаю, что это перспективное направление.

— Виктор Васильевич, в этой связи вопрос: как лично Вы относитесь к сотрудничеству с кооперативами, хозрасчетными объединениями, совместными предприятиями? Могут ли они решить проблемы в этой области?

— Я бы поставил вопрос так: могут ли кооператоры решить для нас проблему создания универсальных программных инструментальных средств для автоматизации процесса разработки конкретных прикладных программ самим педагогом? Наверное, могут, но эту политику нам будет осуществлять очень сложно. Мне думается, что эти вопросы должны все-таки решаться скоординированной государственной политикой, созданием целевых временных творческих коллективов, с привлечением всех сил, которые уже имеются на сегодняшний день.

Я считаю, что здесь нам нужна стратегия и концепция использования ВТ и программных средств. На ее основе должна быть разработана такая технология создания этих средств, которая могла бы обеспечить разработку педагогом необходимых фрагментов



учебных курсов в логике своей методики обучения. Педагогам нужны общие методики формирования программных средств.

Третье подразделение — *отдел научно-организационного управления Комплексной программы научно-технического прогресса стран-членов СЭВ*. Здесь мы очень надеемся, что сумеем более эффективно использовать опыт наших коллег-соисполнителей из социалистических стран. Мы рассчитываем, что удастся организовать творческие совместные коллективы, которые будут заниматься универсализацией и типизацией разработки инструментальных и технических средств ВТ, а также вопросами методики применения программных средств.

У нас есть большой *отдел методики профессионального обучения*. Мы хотим его полностью сориентировать на решение проблемы соединения новых информационных технологий с оправдавшими себя традиционными технологиями обучения и преподавания. С другой стороны, мы считаем, что если не затронем существо методики обучения и будем привносить новые информационные технологии в виде отдельных фрагментов, то существенного эффекта получить не сможем. Это очень глубокая проблема.

Таким образом, у нас практически половина института будет заниматься вопросами использования новых информационных технологий в системе профтехобразования.

— Коррелируют ли исследования, проводящиеся в вашем институте, с аналогичными в области школьного образования? Я знаю, что институт сотрудничает и с АПН СССР. Насколько вы открыты для этого

сотрудничества или все-таки приходится идти своим путем?

— Я бы сказал, что на уровне человеческих контактов, в плане научных интересов и взаимного обмена у нас все в порядке. Но если говорить об организационно-плановых вопросах, мне кажется, что все эти достаточно мощные научные силы были ориентированы на школьную тематику. Это касается и ИПИ АН СССР в смысле тех структурных подразделений, которые ему разрешили открыть, и ИК АН УССР, получившего целевое задание, и недавно созданного НИИ ИВТ АПН СССР. В своих программах они предусмотрели исследования по вопросам компьютеризации разных уровней образования, но реально каких-либо проблем, принципиально связанных со спецификой профтехшколы, пока не решили. Возможно, здесь не хватило и нашей инициативности, хотя у нас есть и договоры, и отдельные очень интересные совместные работы. Но гораздо больше еще только предстоит сделать, и ждать у нас времени нет.

С другой стороны я вижу, что взаимопонимание сейчас наблюдается на всех уровнях, и мы надеемся, что Госкомитет СССР по народному образованию, говоря сегодня о единой стратегии внедрения информационных технологий, наше место, нашу экологическую нишу в этом процессе видит и это будет способствовать нашему развитию достаточно эффективно.

— Большое спасибо за беседу. Мы желаем Вам творческих успехов.

Беседу провела А. Кравцова

## Экспериментальное хозрасчетное творческое объединение

### «ЛЕФОРТОВО»

предлагает комплект поурочных программ для КУВТ-86 (Бейсик) для обучения курсу информатики в IX—X классах (всего 91 программа).

Информация записана на гибких магнитных дисках исполнителя.

*Справки по адресу:*

*101000, Москва, ул. Чернышевского, 10, строение 3,*

*ЭХТО «Лефортово»,*

*подразделение 22. СПЕКТР.*

*С. А. Короткову.*

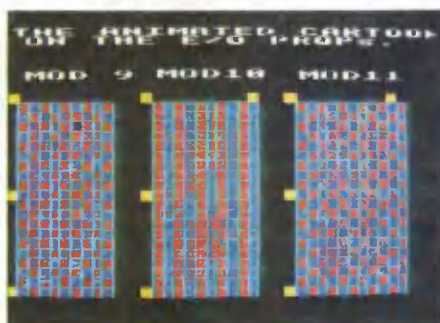
Нашим читателям уже знакомо имя А. А. Зенкина (см. статью «Обучающая система-тренажер» в «ИНФО» № 5 за 1988 г.). Сегодня мы расскажем о работе, которую он считает главным своим свершением. Это создание нового класса компьютерных программ, основанных на использовании так называемой когнитивной интерактивной компьютерной графики (ККГ).

## Экология мышления

Что такое ККГ? Этой аббревиатурой А. А. Зенкин обозначает подход, ставящий целью изобразить на экране дисплея не форму предметов и явлений, не внешние их проявления, а содержание, внутреннюю сущность. В качестве примера его реализации рассмотрим диалоговую систему для исследования проблем классической теории чисел (ДСТЧ).

Система работает очень просто. Она «нарезает» определенный отрезок числовой оси с заданным шагом (модулем) и составляет из отрезков страницу некоей «кибернетической» книги, которая, как и обычная, читается слева направо и сверху вниз. На экране дисплея числам соответствуют квадратики (фото 1). Некоторые числа выделяются цветом: на фото 2 выделена «чет-нечетная» структура натурального ряда (красные квадратики — нечетные числа, синие — четные) в трех «разрезах»: по модулю 9, 10 и 11. Научная ценность этого построения, конечно, невелика — на нем видно лишь, что четные и нечетные числа строго чередуются.

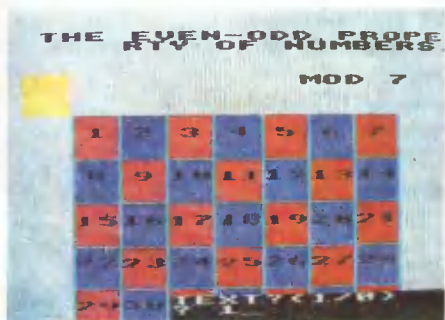
На фото 3 красные квадратики соответствуют простым числам. Их упорядоченность

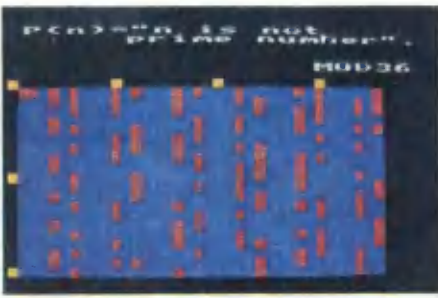


уже более интересна. Числовая ось здесь «нарезана» по модулю 35.

На фото 4 — те же простые числа, но уже по модулю 36. Картина резко изменилась, бросается в глаза упорядоченность, порожденная правилом: все простые числа, превышающие 3, принадлежат множеству  $\{6n \pm 1; n=1, 2, 3, \dots\}$ .

Но все это мы знали и без ДСТЧ. А можно ли обнаружить среди натуральных чисел 1, 2, 3, ..., 150 нечто, не известное современной науке? Можно ли абстрактнейший раздел фундаментальной науки о числах превратить в захватывающий детек-





состоящего из семи элементов, потребовал нескольких страниц текста. Только применение ИКГ позволило найти следующие множества:  $Z(1, 3)$ ,  $Z(1, 4)$ ,  $Z(1, 5)$ ,  $Z(2, 2)$  —  $Z(10, 2)$  и т. д. На фото 5 — отрезок натурального ряда от 1 до 150, зеленым цветом выделено множество  $Z(1, 3)$ . Чтобы открыть его, пришлось просмотреть 14 кадров по 300 чисел в каждом — понятно, вручную сделать это, не зная, что ищешь, нереально. Зато теперь существует открытый также с помощью ДСТЧ визуальный критерий, ограничивающий возможное число элементов  $Z(m, r)$ . Если бы Г. Полл знал этот критерий, он описал бы  $Z(1, 2)$  тремя строчками текста.

На фото 6 изображено множество  $Z(1, 4)$ . Еще одно открытие, сделанное с помощью ДСТЧ, — набор парабол на фото 7. Здесь

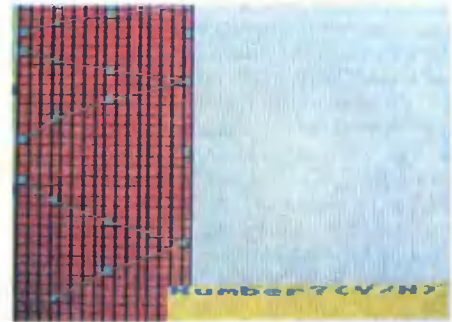
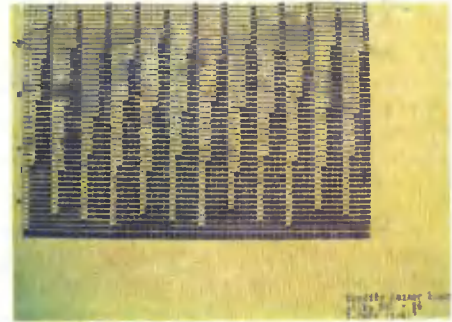
тивно-цветомузыкальный мультфильм? Можно ли, наконец, превратить красоту математических утверждений в конструктивный визуальный критерий поиска научных истин, о котором мечтал великий Анри Пуанкаре, и построить бесконечноэтажный научный небоскреб там, где даже Эйлер и Лагранж, Гаусс и Гильберт видели только фундамент — так называемую проблему Ларинга, знаменитую проблему теории чисел, которая по своей научной значимости котируется среди специалистов не ниже, чем Большая теорема Ферма?

Как вы, читатель, уже догадались, можно. На фото 5 — «автограф» множества  $Z(1, 3)$ , открытого А. А. Зенкиным. Оно принадлежит к классу множеств  $Z(m, r)$  — так в теории чисел обозначают множества натуральных чисел, которые нельзя представить в виде суммы разностей вида

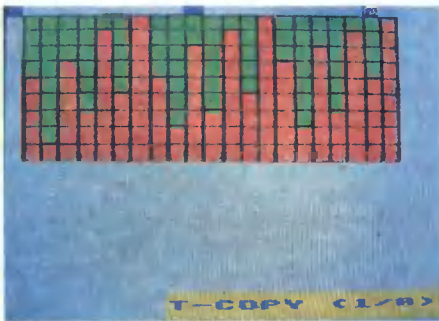
$$\sum_{i=1}^s (n_i - m^2)$$

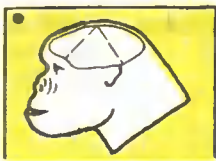
при любых  $S \geq 1$ ,  $n_i \geq m$ .

До недавнего времени в этом классе рассматривались лишь множества  $Z(0, r)$ , которые — увы! — не содержали ни одного элемента. Первый прорыв в исследованиях  $Z(m, r)$  при  $m > 0$  совершил американский математик Годвард Полл. В 1933 г. он выделил множество  $Z(1, 2)$ , но дальше не пошел — трудоемкость задачи чрезвычайно высока и не позволяет сделать следующий шаг, оставаясь в рамках классических подходов и методов. Даже анализ множества  $Z(1, 2)$ ,



белым выделены числа, являющиеся квадратами натуральных чисел (тонкая белая линия прорисована для подчеркивания закономерности). На числовой оси это все более редко расположенные точки 1, 4, 9, 16, 25, ...; если их расположить на плоскости как график функциональной зависимости, получится парабола. На снимке видно, что параболы получаются и в том случае, если «свернуть» числовую ось в двумерную таблицу (в данном случае по модулю 16). Из бесконечной параболы получились бесконечное число конечных парабол — весьма изящное и неожиданное преобразование.





Тут следует отметить, что вывод на экран каждого квадрата в ДСТЧ сопровождается звуком и музыкальные закономерности бывают не менее эффективными, чем визуальные. Увы — в журнале их воспроизвести нельзя.

Теперь, когда вы получили представление о ККГ на примере ее частного случая — ДСТЧ, предлагаем вашему вниманию интервью с А. А. Зенкиным.

— Александр Александрович, почему вы занялись компьютерной графикой? Была ли это игра или вы сразу ставили перед собой какие-то научные цели?

— Я стремился к тому, чтобы с помощью ККГ получить нечто качественно новое. Сегодня хорошо известно, что с помощью математического моделирования, вычислительного эксперимента можно, например, найти оптимальную реализацию системы управления какой-нибудь технологической системой, которая будет давать на 10—15—20 % больше полезного продукта, повысит надежность, но это всего лишь количественный рост.

— В отличие от, например, открытия «ядерной зимы», тоже сделанного с помощью ЭВМ.

— Согласен, это подходящий пример: не количественное улучшение, а качественный скачок познания, совершенный с помощью вычислительного эксперимента.

Это — общее обоснование. Вторым важным обстоятельством было мое участие в работе по визуализации строения молекул. Именно тогда я увидел, что компьютерная графика (а было это в 1971 г.) дает нечто принципиально новое по сравнению с символьным представлением информации. Мы тогда вывели на экран обычного бытового телевизора довольно большую молекулу циклодекапептида (около 50 атомов). Меня потрясло, что я полгода рассчитывал ее характеристики, свойства, изучил ее, казалось бы, вдоль и поперек, но только увидел изображение на экране, действительно понял, с чем я имею дело.



— Лучше один раз увидеть, чем сто раз услышать.

— Конечно! И ведь все это знают, но многие ли пользуются? А в этой привычной поговорке выражены важнейшие механизмы работы мозга. Его левое и правое полушария не дублируют друг друга (об этом можно прочесть, например, в замечательной книге профессора Д. А. Пospelова «Фантазия или наука»). Левое полушарие преимущественно отвечает за обработку абстрактных, символьных конструкций, правое — за образное мышление. Человеческий разум в целом есть продукт взаимодействия полушарий.

В левом полушарии находится центр речи, реализуется рационально-логическое мышление, упрощенно выражаясь — мышление словами. А в правом реализуется интуитивное мышление. Можно сказать, что там находится центр интуиции, непосредственного постижения.

За последние тысячелетия развитие человеческого мышления и особенно его специфического проявления — науки — было ориентировано на левое полушарие. Все достижения, открытия формулировались с помощью слов, формул и т. п.; о роли интуиции в свершении этих открытий не говорили, ее маскировали...

Как готовят к научной деятельности? Сегодняшнее образование — это почти исключительно вербальное воздействие на мозг студента. Лекции на 99 % состоят из слов, и даже если на доске рисуется схема, то только для того, чтобы объяснить ее словами, полагая, что образ — это недостаточно строго.

Собственно, схемы-то эти, как правило, не являются образами, а только лишь графическими выражениями вербализованной абстракции.

Что такое сегодняшнее воспитание, начиная с детского сада?

— Уговаривание.

— Слова, слова! Нотации, «внушения», которые ничего не внушают, а только лишь декларируют... Ясли, школа, вуз — все ориентировано на левое полушарие. Правое игнорируется, как будто его нет. Оно атрофируется!

— Ну, природа-то достаточно активно воздействует на него, так что атрофии бояться не стоит.

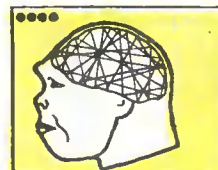
— Природа, конечно, берет свое, но достаточно ли человека пренебрегать половиной мозга? Даже больше, чем половиной, так как взаимодействие левого и правого полушарий дает новое качество. Именно образное мышление, интуиция — основа творческого, в том числе и научного, мышления.

Кажется, что величайшими достижениями современная наука обязана левому полушарию, но это заблуждение порождается тем, что мы не детализируем процесс появления нового знания. Если же взглядеться... Великий математик Пойа отметил, что прежде чем доказать, нужно догадаться, что доказывать. А Эйнштейн сказал, что ни один ученый не мыслит символами, т. е. словами. И ему можно верить — он-то был ученым и знал толк в открытиях. Вспомните его ответ на вопрос сына о причине его известности. Он сказал: «Представь жука, ползущего по очень большому шару. Жук думает, что он ползет по плоскости. А я был первый, кто заметил, что это не плоскость, а шар». Это сугубо правополушарный образ, очень многое объясняющий.

Вот мой основной постулат: мы имеем богатейший, тысячелетний, опыт воздействия с помощью слова на образ мысли человека, на левополушарное мышление, на поведение. По аналогии: почему бы с помощью специально выбранного образа не попытаться воздействовать на правое полушарие с тем, чтобы если не породить определенные мысли (сейчас поставить такую задачу было бы слишком смело), то хотя бы инициировать деятельность мышления в определенном направлении? Воздействовать на живое созерцание с помощью образов! И вот, отталкиваясь от этой, на мой взгляд, очень здоровой, идеи, я и попробовал реализовать это целенаправленное воздействие с помощью образов, причем образы строятся с помощью компьютера.

ЭВМ активно используется для построения изображений реальных объектов, явлений, процессов. Следующий шаг — выход в мир абстрактных понятий. Почему в этом возникает необходимость? Телескоп создан, чтобы видеть удаленные объекты, микроскоп — для разглядывания малых. Тепловизор позволяет видеть инфракрасное излучение. Но все это — разглядывание материи. А ведь наука есть идеальное отражение этой реальности, причем отражение на языке абстрактных понятий. Существует два мира: материальный и идеальный. Пора создавать технические средства, позволяющие заглянуть, именно заглянуть, в идеальный мир. ККГ — именно такое средство.

Конечно, графика сейчас очень широко используется в САПР и математическом моделировании, для представления числовой ин-



формации в виде графиков, гистограмм и т. п. Здесь тоже есть элемент когнитивности. Но как иллюстративная графика занимает «пространство» от черно-белой схемы до точного цветного изображения, так и когнитивная графика — от попытки отображения внутренних количественных связей на уровне графиков до отображения теоретических абстракций, сути внутреннего содержания проблемных областей.

Сегодня мы стоим на пороге новой технологии познания, связанной с переходом от переработки линейной буквенно-цифровой информации к принципиально многомерной, многоканальной (звук — цвет — образ...) форме.

Когда-то живое созерцание и абстрактное мышление шли рядом. Леонардо да Винчи — художник, скульптор, поэт, инженер, изобретатель, ученый, конструктор... После XVII в., когда начался могучий рывок научного познания, происходит отрыв — абстрактное мышление все дальше уходит от живого созерцания. Одна из бед не только современной науки, но и всей цивилизации (даже искусство становится все более рассудочным) — катастрофический разрыв абстрактного мышления и живого созерцания, или между левым и правым полушариями.

— Правое полушарие оказалось дискриминированным.

— Да, даже в социальном смысле. Сегодня появляются альтернативы в оценке всевозможных событий, ситуаций, явлений и прочего — но нет альтернативы научному мышлению, и чем оно абстрактней, непонятней, тем больший вызывает пиетет. Человек не рассуждает, когда ему говорят, что это сказали ученые. Хотя мы знаем, что ученые могут и заблуждаться, и лгать. Писателям, выступающим с экологическими высказываниями, постоянно бросают упреки в дилетантизме. А ведь это следствие того, что нарушена «экология мышления», презирается его образный компонент.

Итак, разрыв абстрактного мышления и живого созерцания является причиной огромного числа бед современной цивилизации в целом. Возьмем образование. В «Коммунисте» (№ 12 за 1988 г.) напечатана статья В. Владимирова и Л. Фаддеева о современном математическом образовании. Они пишут, что математика теряет привлекательность в глазах современной молодежи;



связано это с тем, что она становится все более абстрактной. По их мнению, чтобы пробудить к математике интерес, главные усилия нужно бросить на популяризацию. А что такое популяризация? Представление научных знаний в такой форме, чтобы они стали доступны для образного восприятия неспециалиста, школьника, студента и т. п. С формальной точки зрения, это ни что иное, как низведение абстрактного мышления до уровня живого созерцания. Низведение в хорошем смысле. Ведь это, во-первых, объективная необходимость. А во-вторых, есть множество исторических примеров, когда именно популярная книга превращала человека в ученого. Так было, например, с Ньютоном, который в 22 года купил популярную книгу по математике, проштудировал ее — и с этого момента начался известный нам Ньютон. Эйнштейн в автобиографии несколько раз подчеркивает, что его становлению как ученого очень помогли популярные изложения классических физических теорий.

«Популяризаторской» можно назвать и функцию ККГ, правда, уже на принципиально новом уровне: она позволяет ПОДНЯТЬ живое созерцание до абстрактного мышления любого уровня. Я убедился в этом с помощью ДСТЧ. Такими вещами занимаются на математических факультетах, на спецкурсах, не ранее шестого семестра. А представленные картинками позволяет понимать эту тему уже пятиклассникам. Сначала они, естественно, бывают просто заинтересованы картинками. Но стоит объяснить им, что означают эти цвета, эти регулярности, каков их математический смысл, в детях просыпается какой-то охотничий инстинкт: они начинают самостоятельно искать закономерности орнамента, начинают сами строить гипотезы, и без какого-либо «обучающего воздействия», нудного объяснения они достаточно квалифицированно формулировали суть проблемы (конечно, на собственном им, нестрогом, языке) — она становилась им понятна. ККГ — мощнейшее средство познания и обучения, даже сейчас, когда сделан только первый шаг.

Что нас ждет в будущем, бессмысленно даже гадать. Лет через 50 мы будем смотреть на сегодняшний день так, как сейчас смотрим на времена Пифагора. Я это говорю, базируясь на своем опыте. При всем критическом к нему отношении я убежден, что

через 15—20 лет мы не узнаем теории чисел. Изменяются акценты: традиционные задачи, видимо, отступят на второй план, а взамен возникнут принципиально новые классы задач — просто потому, что ЭВМ открывает огромные новые области приложения сил. А ведь натуральные числа — это базис теории множеств, всей математики...

— Возможно, ЭВМ сами «научатся» развивать теорию чисел?

— Скорее, наоборот. Не знаю, каким именно путем будет развиваться «интеллектуальное» применение ЭВМ, но, полагаю, не только тем, который освоен за последние десятилетия. Известные до сих пор попытки создания искусственного интеллекта (ИИ) — это стремление вложить человеческий способ мышления, в основном левополушарный, в какой-то объект (ЭВМ или другой), т. е. как бы отделить мышление от человека и передать соответствующий алгоритм автомату. Со строго формальной точки зрения, наверное, только такая конструкция и может считаться искусственным интеллектом, интеллектом, мыслящим вне и независимо от человека. Я думаю, однако, что вполне правомерным является альтернативный подход, ориентированный на симбиоз человека и машины. Зачем их разделять? Для удовлетворения научного любопытства? Пожалуйста! Конечно, такие эксперименты интересны и могут стать полезными. Но это разделение не является необходимым и пока не оправдывает себя с практической точки зрения. Так давайте развивать направление, позволяющее человеку и машине действовать в рамках единой системы и тем самым на порядок усиливать потенциал естественного интеллекта. Ведущий специалист по ИИ академик Г. С. Поспелов считает, что будущее интеллектуальных систем не в противопоставлении искусственного разума машины естественному человеческому, а в их взаимодополнении.

Еще раз подчеркиваю: я имею в виду взаимодействие, в котором машина играет роль не поставщика формально-логической информации для левого полушария, а возбудителя правого полушария и поставщика образной информации для него.

Подавляющее большинство современных систем ИИ предназначены для переработки входной информации в выходную (сбор, хранение и переработка, как правило, на уровне перекодировки, сортировки, сжатия, реферирования). А предназначение системы ДСТЧ — помочь породить знания. Она строит образ, отражающий суть данной предметной области, этот образ воздействует на правое полушарие человека и активизирует в его мозгу творческие процессы. Тут

существенно, что именно под воздействием ККГ рождается новая истина. Экспериментальное доказательство — тот факт, что без применения ДСТЧ эти истины (например, касающиеся множеств  $Z(m, r)$ ) доказаны не были. Итак, первая принципиальная особенность ДСТЧ в том, что это *знаниепорождающая система*.

Ее вторая принципиальная особенность в том, что она порождает новые знания *не вместо, а вместе* с человеком. Она не моделирует творческие способности человека, а усиливает их.

Третья особенность в том, что ДСТЧ обеспечивает общение человека не с машиной, а с *проблемой*, причем именно с сутью проблемы, причем не на так называемом естественном языке, который, конечно, богат и универсален, но условен, слишком далеко отстранен от обозначаемых им явлений, а на языке образов.

— Можно сказать, что это система общения с природой, абстрагированной, очищенной достаточно для выделения конкретной проблемы, природой, препарированной и подготовленной для наиболее эффективного восприятия правым полушарием.

— Абстрагированной и препарированной, н-да... Абстрагированием и препарированием наука занимается тысячи лет. Конечно, без абстрагирования не обойтись, полная, абсолютная истина недостижима. Наука отображает реальность на уровне приблизительного сходства математических моделей, и это все же продуктивней, чем голая описательность, но это неточно, здесь заложено принципиальное несоответствие между природой и нашими представлениями о ней и, следовательно, образом воздействия на нее... Я надеюсь, что выход на визуализацию, на гуманитаризацию науки в противовес современному технократическому мышлению, на правое полушарие позволит нам уменьшить степень абстрагирования, позволит более гармонично взаимодействовать с природой.

— Сейчас это особенно важно — ведь иначе вряд ли удастся решить реально вставшую проблему выживания человечества. В экстремальных ситуациях, когда требуется немедленно принять верное решение, человек полагается на интуицию. И она выручает — если достаточно развита.



— Главное для меня не сами эти картинки, хотя научный эффект от них уже есть. Новое научное знание, полученное с их помощью, — мизер по сравнению с тем, что этот подход может принести потенциально. А ведь ККГ полезна не только в области научной работы; гигантские успехи должно дать ее применение, например, в образовании.

Левополушарная ориентация образования сказалась в пресловутой концепции суммы знаний. Сейчас мы стараемся обучить валу, количеству научных результатов, мы считаем, что уровень образования обеспечивается тем, сколько человек знает из того, что было открыто до него. А вот технологии нахождения нового знания мы фактически не обучаем — собственно, и средств для этого не было. ККГ позволяет сместить акценты в сторону технологии познания. С помощью компьютерных картинок демонстрируется не математическая теорема, а процесс перехода от картинок к теореме. Традиционное рассмотрение доказательства этой теоремы тоже должно присутствовать, но уже не как главный способ обучения придумыванию теорем, а как вспомогательный способ проверки их правильности. В системе ДСТЧ есть режим обучения, содержащий набор сценариев тех открытий, которые были сделаны в рамках этой системы, и когда студент просматривает эти сценарии, он проходит тот же путь, что проходил открыватель этих истин. Так что ДСТЧ — еще и представитель нового класса обучающих систем.

— Что ж, Александр Александрович, остается только пожелать вам успехов.

Материалы подготовил К. Шеховцев

Если вы заинтересовались ККГ, то более подробно сможете прочесть о ней в 1990 г., когда в издательстве «Наука» выйдет книга А. А. Зенкина «Когнитивная компьютерная графика» (в плане издательства она фигурирует под названием «Интерактивная компьютерная графика»).

## По страницам тематических планов издательств на 1990 г.

1. Бейсик в примерах и задачах.— М.: Наука, 1990 (II кв.).— 10 л.— (Кибернетика: неограниченные возможности и возможные ограничения).— 1 р. 30 к., 100 000 экз.

Знакомство с миром персональных компьютеров путем решения занимательных задач. Устройство ПЭВМ и основные приемы программирования на Бейсике. Рассчитана на широкий круг читателей.

2. Бордовский Г. А., Извозчиков В. А., Исаев Ю. В. Информатика в понятиях и терминах: Кн. для учащихся.— М.: Просвещение, 1990 (II кв.).— 15 л.— 65 к., 1 000 000 экз.

Словарь-справочник по информатике, содержащий более 400 терминов, встречающихся при изучении курса информатики в средних учебных заведениях.

3. Будущее искусственного интеллекта.— М.: Наука, 1990 (IV кв.).— 20 л.— 1 р. 90 к., 50 000 экз.

Сборник статей отечественных и зарубежных ученых, в которых рассматриваются разные аспекты искусственного интеллекта и перспективы развития этой отрасли информатики. Для широкого круга читателей.

4. Бусленко В. Н. С компьютером на «ты».— М.: Радио и связь, 1990 (II кв.).— 10,5 л.— (Масовая радиобиблиотека).— 90 к., 100 000 экз.

В доступной форме рассказывается о внедрении информатики в различные области человеческой деятельности, об использовании компьютеров в обучении и играх, современном программировании и эксплуатации информационных ресурсов, перспективах компьютеризации и ее социальных последствиях. Для широкого круга читателей.

5. Вершинин О. Е. За страницами учебника информатики: Кн. для учащихся.— М.: Просвещение, 1990 (IV кв.).— 20 л.— 1 р., 200 000 экз.

Простым языком изложены важнейшие вопросы информатики. Даны исторические сведения, краткие биографические данные о наиболее известных ученых.

6. Дворчик Ш., Василенки Л. Мышка Программышка в стране Информатике: Пер. с франц. М.: Радио и связь, 1990 (I кв.).— 6 л.— 60 к., 100 000 экз.

В увлекательной форме книга знакомит читателей с основными понятиями, необходимыми для игры и работы с персональным компьютером. Содержит разнообразные задания по нарастаю-

щей сложности. Для школьников IV—VI классов и преподавателей информатики.

7. Джонстон Г. Учитесь программировать: Пер. с англ.— М.: Финансы и статистика, 1990 (II кв.).— 27 л.— 2 р., 40 000 экз.

Представляет собой курс лекций профессора К. Хоора по искусству программирования в интерпретации Г. Джонстона. Методика написания программ, выбора алгоритмов, решения задач иллюстрируется программами минимальной сложности, реализованными на языке Паскаль. Предназначена для специалистов, преподавателей и студентов вузов.

8. Дудкин В. Д., Чертков Б. М., Ицкович А. Ф. Основы эксплуатации персональных компьютеров: Практическое пособие. М.: Высшая школа (вузы и техникумы), 1990 (III кв.).— 15 л.— 80 к., 100 000 экз.

Рассматриваются персональные компьютеры и учебные вычислительные комплексы на их основе, в первую очередь ДВК и БК; языки программирования Бейсик, Фокал; перспективы развития учебной вычислительной техники. Рассчитана на широкий круг читателей.

9. Дьюдни А. К. Занимательный компьютер: Пер. с англ. М.: Мир, 1990 (II кв.).— 25 л.

Иллюстрированная научно-популярная книга, составленная из статей, опубликованных в журнале «В мире науки» в период 1984—1988 гг.

10. Егоров Ю. Н., Голубев Н. А. Уроки робототехники. М.: Радио и связь, 1990 (II кв.).— 4 л.— (Науч.-поп. б-ка школьника).— 75 к., 20 000 экз.

В популярной форме рассказывается об истоках возникновения робототехники, устройстве роботов и их основных узлов (в первую очередь электронных органов управления — процессоров, запоминающих устройств, датчиков и т. д.). Для старшеклассников.

11. Зарецкий А. В., Труханов А. В. А я был в компьютерном городе: Кн. для учащихся.— М.: Просвещение, 1990 (III кв.).— 12 л.— 1 р., 300 000 экз.

Первый опыт синтеза сказки-детектива и научно-популярной энциклопедии для малышей в картинках. Авторы серьезно, понятно и доступно рассказывают младшим школьникам о дискеетах и дисплеях, о машинном языке и программировании.

12. Зуев К. А. Компьютер и общество. М.: Политиздат, 1990 (I кв.).— 10 л.— 50 к.



Книга затрагивает актуальные социальные проблемы компьютерной революции.

13. Ильин В. П. Вычислительная информатика: Открытие науки. Новосибирск: Наука, 1990 (IV кв.).— (Наука и технический прогресс).— 85 к., 15 000 экз.

Книга знакомит читателя с феноменом компьютеризации человеческой деятельности, рассказывает о становлении новой науки, возникшей на стыке вычислительной техники, программирования и вычислительной математики, о задачах и принципах математического моделирования процессов и явлений.

14. Информатика: Учеб. пособие для студентов пед. ин-тов/ А. Р. Есаев, В. И. Ефимов, Э. А. Пашенко и др.— М.: Просвещение, 1990 (III кв.).— 13 л.— 65 к.

Даются описание языка Бейсик, модели, методы, алгоритмы и программы обработки информации и решения задач; приведены примеры педагогических программных средств.

15. Кергалл М. Методы программирования на Бейсике (с упражнениями): Пер. с франц.— М.: Мир, 1990 (III кв.).— 17 л.— 2 р. 20 к., 50 000 экз.  
В книге 120 упражнений, часть из них с комментариями и ответами.

16. Компьютер обретает разум: Пер. с англ.— М.: Мир, 1990 (II кв.).— 35 л.— 5 р., 100 000 экз.

Научно-популярная книга об основах компьютерной графики и принципах построения систем с искусственным интеллектом.

17. Компьютеры и познание (очерки по когнитологии).— М.: Наука, 1990 (II кв.).— 20 л.— (Кибернетика: неограниченные возможности и возможные ограничения).— 1 р., 50 000 экз.

Сборник знакомит читателя с перспективами когнитивной науки, возникшей и стремительно развивающейся на стыке информатики, психологии, философии, лингвистики и физиологии. В центре ее внимания — разнообразные формы представления знаний, взаимодействие человека и компьютера, организация познавательных процессов и архитектура вычислительных устройств. Рассчитана на самую широкую аудиторию.

18. Кузнецов Ю. К., Котова Л. И., Исаев Ю. В. Вычислительная техника в учебном процессе: Учеб. пособие для учащихся пед. училищ.— М.: Просвещение, 1990 (III кв.).— 8,5 л.— 30 к.

Пособие соответствует курсу ОИВТ для педучилищ.

19. Логика и компьютер.— М.: Наука, 1990 (IV кв.).— 10 л.— (Кибернетика: неограниченные возможности и возможные ограничения).— 70 к., 5 000 экз.

Впервые в отечественной научно-популярной литературе детально анализируются логические предпосылки информатики, показывается ключевая роль логики в программировании, в разработке экспертных систем, в исследованиях по искусственному интеллекту.

20. Мартузан Б. Я. Подружись со мной, компьютер: Кн. для учащихся.— М.: Просвещение, 1990 (III кв.).— 14 л.— 65 к., 200 000 экз.

Автор в занимательной форме излагает основные понятия и приемы программирования. За основу взят язык Бейсик для СМ-4, ДВК-2 и «Корвета».

21. Наш друг — компьютер: Рек. библиогр. указ.

(Гос. б-ка СССР им. В. И. Ленина / Сост. Н. В. Черемушкина, Л. М. Кузнецова, З. П. Джинова).— М.: Кн. палата, 1990 (II кв.).— 8 л.— 35 к., 40 000 экз.

Рекомендуется наиболее интересная литература по актуальным проблемам компьютеризации.

22. Новичков В. С., Парфилова Н. И., Пылькин А. Н. Паскаль: Учеб. пособие.— М.: Высшая школа (вузы и техникумы), 1990 (I кв.).— 10 л.— 30 к.

Для учащихся техникумов, изучающих ОИВТ.

23. Очков В. Ф., Пухначев Ю. В. 108 советов начинающему программисту.— М.: Энергоатомиздат, 1990 (I кв.).— 17 л.— 2 р. 10 к., 200 000 экз.

Приведены и разобраны программы решения различных задач на персональном компьютере.

24. Перегудов М. А. Компьютер-редактор.— М.: Финансы и статистика, 1990 (III кв.).— 10 л.— (Человек и компьютер).— 75 к., 50 000 экз.

В коротких рассказах, главным «действующим лицом» которых является компьютер, в понятной форме излагаются возможности современной ВТ при работе с текстовой информацией.

25. Персональный компьютер с разных сторон.— М.: Наука, 1990 (III кв.).— 1 р., 50 000 экз.

Освещаются наиболее интересные стороны инженерно-технического обеспечения современных персональных компьютеров.

26. Поддубная Т. Н., Фукс И. Л. Информатика в задачах и упражнениях.— Томск: Радио и связь, 1990 (I кв.).— 9 л.— (Массовая радиобиблиотека).— 70 к. 125 000 экз.

На основе всестороннего и детального анализа ряда практических задач изложены начала информатики. Решение задач доведено до готовых программ на языках Паскаль и Бейсик. Учащимся и преподавателям ПТУ, студентам и преподавателям вузов.

27. Русецкий А. Ю. В мире роботов: Кн. для учащихся.— М.: Просвещение, 1990 (III кв.).— 10 л.— 80 к., 100 000 экз.

Автор знакомит читателей с историей робототехники и перспективами роботостроения.

28. Современное производство и компьютер.— М.: Наука, 1990 (IV кв.).— 10 л.— (Кибернетика: неограниченные возможности и возможные ограничения).— 70 к., 50 000 экз.

Анализируются проблемы и перспективы современной компьютерной революции, ее воздействие на развитие техники и производства.

29. Сурин А. В. Компьютеризация в современном мире (мировоззрение, методология, социальные проблемы) — М.: МГУ, 1990 (IV кв.).— 15 л.— 1 р. 20 к.

Анализ компьютеризации как целостного и неоднозначного социокультурного явления. Влияние компьютеризации на характер труда, ее роль в развитии современной науки. Трудности и противоречия компьютеризации образования.

30. Урнов В. А., Климов Д. Ю. Преподавание информатики в компьютерном классе: Кн. для учителя.— М.: Просвещение, 1990 (I кв.).— 14 л.— 60 к., 200 000 экз.

Методическое руководство по преподаванию курса ОИВТ в IX—X классах (машинный вариант) на базе ДВК-2, КУВТ-86, ДВК-3 и УКНЦ. Книга содержит большое количество задач, упражнений и примеров.

31. Учебный терминологический словарь по основам информатики и вычислительной техники: Пособие для учащихся / А. П. Ершов, Н. М. Шанский, А. П. Окунева, Н. В. Баско.— М.: Просвещение, 1990.— 70 к., 500 000 экз.

32. Халамайзер А. Я., Перегудов М. А. Ключ к компьютеру.— М.: Радио и связь, 1990 (III кв.).— 14,5 л.— (Массовая радиобиблиотека).— 1 р. 20 к., 100 000 экз.

В популярной форме рассказывается о роли программных средств вычислительной техники в системах информатики.

33. Черемных С. В., Поляк Ю. В. Диалог с персональным компьютером.— М.: Финансы и статистика, 1990 (III кв.).— 12 л.— (Человек и компьютер).— 85 к., 50 000 экз.

Задача авторов — помочь пользователям разной квалификации и специалистам сориентироваться среди обилия технических и программных средств.

34. Чернов Б. И. Программирование на алгоритмических языках Бейсик, Фортран, Паскаль: Кн. для учащихся.— М.: Просвещение, 1990 (IV кв.).— 11 л.— (Мир знаний).— 60 к., 200 000 экз.

35. Школа и новые информационные технологии: М.: Наука, 1990 (IV кв.).— 10 л. (Кибернетика: неограниченные возможности и возможные ограничения).— 1 р., 70 000 экз.

36. Яковлев А. Компьютер в школе.— М.: Педагогика, 1990 (III кв.).— 10 л.— 55 к., 100 000 экз.

Цель книги — рассказать о возможностях использования компьютерной техники в учебном процессе, оказать помощь учителю в организации работы школьников с компьютерами. Рассчитана на учителей.

37. Ямпольский В. С. Основы автоматизации и электронно-вычислительной техники: Учеб. пособие для студентов пед. ин-тов.— М.: Просвещение, 1990 (II кв.).— 14 л.— 70 к.

Пособие написано в соответствии с программой нового курса физ.-мат. факультетов пединститутов «Основы автоматизации и ЭВТ».

38. Толковый словарь по информатике: Справ. изд. / В. И. Першиков, В. М. Савинов.— М.: Финансы и статистика, 1990 (III кв.).— 32 л.— 2 р. 20 к., 60 000 экз.

Содержит около 9 тыс. терминов по информатике, техническим средствам, программному обеспечению, системам обработки данных. Особое внимание уделяется терминологии в новых областях информатики (ПЭВМ, базы данных, распределенная обработка, искусственный интеллект).

39. Осваиваем микрокомпьютер: В 2 т.: Пер. с англ. / Дж. Тэтчел, Б. Беннетт, К. Фрейзер и др.— М.: Мир, 1990 (IV кв.).— 26 л.— 4 р. 60 к., 125 000 экз.

Описаны устройство ЭВМ, работа ее узлов и области применения. Много внимания уделено машинной обработке текстовой информации. Существенную часть составляет практикум по Бейсику.

40. Коутс Р., Влеминкс И. Интерфейс «человек — компьютер»: Пер. с англ.— М.: Мир, 1990 (IV кв.).— 26 л.— 2 р. 20 к., 40 000 экз.

Книга посвящена разработке программного обеспечения, представляющего пользователю эффективные средства общения с ЭВМ.

41. Нестеренко А. В. ЭВМ и профессия программиста: Кн. для учащихся.— М.: Просвещение, 1990 (IV кв.).— 12 л.— 1 р., 150 000 экз.

*Сост. В. Фрейман*

## Уважаемые авторы!

При подготовке статей, направляемых в редакцию «ИНФО», просим учитывать следующее.

Рукопись должна быть отпечатана на пишущей машинке в двух экземплярах через два интервала на стандартных листах писчей бумаги.

Рукописи, представленные в виде принтерных распечаток, принимаются в работу только, если текст выполнен на белой (обычного формата) бумаге и достаточно контрастен, размер шрифта близок к стандартному, расстояние между строками не менее двух интервалов и заглавные буквы отличаются размером от строчных.

Прилагаемые эскизы, схемы, таблицы должны быть выполнены в двух экземплярах, четко и аккуратно, снабжены необходимыми пояснениями.

Черно-белые фотографии должны быть отпечатаны на глянцевой бумаге размером не меньше 9×12 см, без повреждений, изломов, царапин. Пояснительные надписи выполняются на оборотной стороне мягким карандашом. Принимаются качественные (желательно широкоформатные) цветные слайды.

На последней странице рукописи необходимо указать сведения об авторах: фамилию, имя, отчество, домашний адрес, место работы, должность, количество детей, телефон.

Рукописи, не отвечающие перечисленным требованиям, рассматриваться не будут.

# ВНИМАНИЮ ПРЕДПРИЯТИЙ И ОТДЕЛЬНЫХ ГРАЖДАН!

**Если вы** намерены приобрести персональный компьютер для предприятия или в личное пользование, хотите максимально эффективно использовать компьютерную технику на своем предприятии, едете в длительную командировку за рубеж, где вам неизбежно придется иметь дело с персональными компьютерами,  
Научно-производственное объединение «Импульс»

ИПК Министерства связи СССР

## ГОТОВО ОКАЗАТЬ ВАМ ПОМОЩЬ В ОБУЧЕНИИ РАБОТЕ НА ПРОФЕССИОНАЛЬНЫХ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРАХ.

В процессе обучения слушатели получают необходимые навыки для использования самых современных моделей персональных компьютеров в одной из массовых профессий: инженер, бухгалтер, экономист, секретарь и т. д.

Это дает возможность выполнять работу, требующую высокой квалификации, с использованием компьютерной техники как в СССР, так и за рубежом.

**Обучение организуется как для тех, кто ранее не работал с компьютером, так и для подготовленных пользователей.**

**В процессе обучения начинающие осваивают работу на персональном компьютере и прикладные программы:**

- редактор текстов (VWrite); ● базу данных (dBASE III);
- операционную систему (MS DOS); ● электронные таблицы (Supercalk);
- основы языка Бейсик и другие программы, необходимые для практической работы.

**Более подготовленные слушатели могут изучить:**

- язык Си; ● язык Турбо-Паскаль;
- Ассемблер-86; ● пакеты программ типа Auto Cad.

При необходимости объем и содержание учебного курса могут быть изменены.

По желанию слушателей организуется индивидуальное обучение, а также консультации.

Обучение проводят преподаватели высшей квалификации — ведущие специалисты отрасли, имеющие опыт использования в практической работе современной компьютерной техники.

Обучение организовано на ПЭВМ, совместимых с наиболее распространенными и перспективными компьютерами IBM PC, ставшими мировым стандартом.

Иногородним слушателям, в том числе обучающимся индивидуально, предоставляется жилье. В ряде случаев обучение может проводиться непосредственно на предприятии заказчика

с отрывом или без отрыва от производства, в том числе за пределами Киева.

**Срок обучения — 1,5—2 месяца. Занятия начинаются ежемесячно.**

**Стоимость — в зависимости от выбранного курса и продолжительности обучения.**

**Для индивидуальных пользователей установлена скидка 50 %.**

**Наш адрес: 252065, Киев, ул. Метростроевская, 5а, НПО «Импульс».**

**Телефоны для справок: 444-45-50, 488-96-46, 488-38-66.**

## Внимание владельцев персональных компьютеров БК-0010

Научно-производственное объединение «ИМПУЛЬС»  
ИПК Минсвязи СССР предоставляет  
индивидуальным пользователям, школам, профтехучилищам,  
предприятиям, организациям, кооперативам

### **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ КОМПЬЮТЕРА БК-0010.**

Объединение располагает фондом из более чем 1800 программ,  
в том числе свыше 700 — в машинных кодах.

Программы записываются на кассеты заказчиков,  
которые необходимо высылать вместе с заявкой.

Заказ может производиться как на конкретные программы по каталогу,  
так и на определенную тематику.

Кроме того, во временное пользование для перезаписи  
предоставляются готовые тематические подборки  
и комплекты лучших, избранных программ  
на кассетах объединения (с последующим возвратом).

В частности:

*пакет на 5 кассетах «Все игры в машинных кодах»;*

*пакеты «25 и 50 лучших игр в кодах»;*

*«Программы на Т-языке»;*

*«Компьютерная графика»;*

*«Компьютерная музыка и речь»;*

*«Психологические тесты и биоритмы»;*

*пакет лучших прикладных программ «Минимакс»;*

*пакет лучших прикладных и игровых программ  
для БК «Основной комплект» и т. д.*

Заказчикам предоставляются также комплекты «Лучшие новинки»,  
в которые входят 15—20 программ ежемесячно.

Объединением подготовлен «Каталог программного обеспечения БК-0010»  
с указанием содержания и полных данных о 1200 программах.

Рассылается в порядке очередности. Стоимость — 15 руб.

Ежемесячно заказчикам рассылается информационный бюллетень  
о новых программах и новостях компьютерной жизни.

Объединение предоставляет индивидуальные консультации  
по вопросам программирования и использования БК —  
как письменно, так и по телефону.

Для школ, предприятий, кооперативов объединение производит  
разработку программ, в том числе учебных,  
а также доработку уже существующих программ.

Расценки на программы для индивидуальных пользователей —  
от 1 до 5 руб.

Оплата заказов производится после их выполнения,  
наложенным платежом.

Для предприятий и организаций  
заказы выполняются по безналичному расчету.

Подробная информация об услугах,  
предоставляемых объединением,  
высылается по первому требованию.

**Наш адрес: 252053, Киев-53, а/я 29, НПО «Импульс». Тел. 444-45-50.**

## Всесоюзная конференция в Севастополе

В июне 1989 г. состоялась Всесоюзная научно-практическая конференция «Компьютерная технология обучения в высшей школе», организованная Севастопольским филиалом республиканского Дома экономической и научно-технической пропаганды общества «Знание» Украинской ССР, ВНИИ проблем высшей школы, Институтом кибернетики им. В. М. Глушкова АН УССР, Севастопольским высшим военно-морским инженерным училищем, Севастопольским приборостроительным институтом, Крымским областным правлением ВНТО им. акад. А. Н. Крылова

Конференция проводилась с целью уточнения тенденций развития компьютерной технологии обучения в высшей школе, обмена опытом создания и применения инструментальных и прикладных средств компьютерной технологии обучения (КТО) и организационно проходила в рамках пленарных и трех секционных заседаний.

В конференции приняли участие 288 представителей 172 НИИ, вузов и других организаций из 35 городов страны. Всего было заслушано 85 докладов (в числе выступавших — 7 докторов и 59 кандидатов наук).

Конференция началась с пленарного заседания, на котором с докладом «Компьютерная технология обучения. Современное состояние и перспективы развития» выступил председатель оргкомитета конференции профессор А. М. Довгялло. «Адаптация в процессах обучения предметной области» — тема выступления профессора Л. А. Растригина. О перспективных технологиях в образовании на основе обучающихся сред рассказал кандидат психологических наук С. Ф. Сергеев (г. Тула).

На секции «Методическое и психологичес-

кое обеспечение компьютерной технологии обучения» большой интерес вызвали доклады А. А. Жаркова, Г. Е. Корсакова, С. Н. Павлова (г. Томск), связанные с подходами к построению интеллектуальных обучающих систем на основе интеграции экспертных обучающих систем, видеотехники и распределенных баз данных и знаний. Е. В. Васильева (Москва) поделилась опытом решения психолого-педагогических проблем создания автоматизированных учебных курсов. Н. В. Грунтович (г. Севастополь) изложил методики разработки обучающей программы с элементами адаптации.

На секции «Программное обеспечение учебных курсов» особый интерес вызвали доклады В. А. Петрушина, В. А. Третьяка (Киев), в которых авторы изложили проект инструментария создания экспертных обучающих программ на базе ПЭВМ. Подход, предложенный в проекте, выгодно отличается разработкой специального инструментария создания обучающих программ, а не адаптацией существующих «оболочек» экспертных систем для целей обучения.

На секции «Использование персональных ЭВМ в учебном процессе вузов» выступил доктор технических наук В. Н. Захаров (Москва) с докладом «Системы программного обеспечения практических занятий по решению вычислительных задач на ПЭВМ», в котором изложил требования к обучению специалиста в профессионально ориентированных вычислительных средах. Е. В. Сурменов (г. Энгельс) поделился опытом подготовки автоматизированных обучающих курсов для ПЭВМ. В. И. Ростокин (Москва) рассказал о модельных экспериментах на ПЭВМ в курсе общей физики.

Особый интерес вызвали доклады и де-

монстрации сотрудников СВВМИУ, где работы по КТО ведутся не отдельными энтузиастами, а комплексно, с участием руководства, преподавателей, научных сотрудников, учебно-вспомогательного персонала и курсантов. За короткий срок в СВВМИУ достигнуты поразительные успехи во внедрении и адаптации существующих систем, а также в разработке собственных программ для основных типов ЭВМ: ЕС, СМ и ПЭВМ.

В дискуссию вылилось обсуждение вопроса о том, для какого вида обучения применять ЭВМ. Большинство предложений сводилось к тому, что необходимо обучать умениям и навыкам, а теоретическими знаниями можно овладевать и традиционным способом. Недооценка компьютерного обучения теоретическим знаниям, видимо, связана с тем, что большинство существующих программ ведут его чисто информативным способом. Переписывать обычный учебник в память ЭВМ признано нецелесообразным. Большинство одобрило предложение выбирать по отдельному предмету хороший обычный учебник и реализовывать для него компьютерную поддержку в виде набора демонстрационных программ, специализированных компьютерных практикумов, тренажеров, задачиков и т. п.

Однако нам кажется преждевременным снимать задачу компьютерного обучения теоретическими знаниями. В этом отношении многие участники связывают свои надежды с повышением уровня «интеллектуализации» обучающих программ, в первую очередь с применением экспертных систем.

По итогам работы конференции более четко выявлены тенденции развития в высшей школе компьютерной технологии обучения,

инструментальных систем, применяемых для создания обучающих программ, автоматизированных учебных курсов.

Особый интерес участников вызвали работы, связанные с разработкой экспертных обучающих программ и инструментальных систем для их создания. В этом направлении существенных успехов добились представители ИК АН УССР, выполняющие проект по созданию инструментальной экспертной системы на основе баз знаний предметной области, педагога и обучаемого. Интерес проявлен и к инструментальной обучающей системе «Платан» (г. Севастополь), ориентированной на создание программ по технологиям обучающих сред, основанных на ассоциативном вызове изображений, мультиклипов, сообщений и работе с программными окнами.

Участники конференции неоднократно высказывали мнение о слабой координации научно-практических работ в области разработки инструментальных систем автоматизированного обучения и обучающих программ. Подчеркивалось не всегда оправданное увлечение разработкой сценарных АОС, реализующих принципы программированного обучения, тогда как обучающих программ на основе обучающих, игровых сред с использованием моделей изучаемых явлений, объектов, процессов и с применением технологий создания экспертных обучающих программ еще мало.

Единодушно было подчеркнуто, что вузы страны имеют предельно слабую техническую базу компьютеризации обучения, особенно на основе ПЭВМ типа IBM PC. Такое положение может существенно затормозить перестройку образования в высшей школе.

## Уральская летняя школа

У Новосибирской летней школы юных программистов с каждым годом становится все больше последователей во всех регионах страны. Но Урал, где накоплен немалый опыт в области школьной информатики, до последнего времени оставался «белым пятном» на карте ЛШЮП. И вот с 3 по 16 июля 1989 г. в Миассе состоялась первая Уральская летняя школа информационной технологии.

Уже из названия видно, что в программе УЛШИТ, наряду с традиционным программированием, много времени было уделено знакомству с самыми современными ин-

тегрированными пакетами и системами, базами данных, редакторами, комплексами САПР. Три четверти из 240 участников составили старшеклассники из Челябинской, Свердловской, Курганской областей, Башкирии, Красноярска, Иркутска. Им было предоставлено более 60 компьютеров IBM PC, «Агат», ДВК-3, БК-0010, Роботрон-1715, СМ-4 из школ и УПК Миасса.

Были, конечно, и экскурсии, и дискотеки, но главное — занятия. Работали очень интенсивно, несмотря на жару. Но лекцию о компьютерных вирусах слушали на берегу

замечательного озера Тургояк. На входной конференции ребята показали свои домашние заготовки, а на выходной — результаты работы по индивидуальным заданиям. Лучшие были награждены дискетами.

Был проведен очень интересный преподавательский семинар — с демонстрациями и обменом программами, обсуждением новых учебников, спорами до хрипоты. Особенно подробно был разобран «свердловский» учебник, который представила И. И. Данилина из Свердловского пединститута, а миасская учительница Т. М. Алексеева рассказала об опыте использования этого учебника. Большой интерес вызвали рассказы об экспериментальной работе Е. В. Кузнецова (г. Красноярск), В. Н. Агейчика (г. Иркутск), И. П. Фроловой (г. Нижний Тагил), С. С. Хилькевича (г. Даугавпилс). Состоялось продуктивное обсуждение концепции летних школ с представителями новосибирской и красноярской школ А. А. Берсом и

Е. В. Кузнецовым. Специальный курс для миасских учителей прочитала Е. В. Капустина (г. Челябинск).

Организовали школу специалисты Уральского автозавода, КБ машиностроения, миасского филиала Челябинского политеха и челябинские учителя А. П. Погодин и В. Г. Мерзлов. Не все, конечно, вышло идеально. Парадоксально, но факт — на УЛШИТ не было КУВТ «Ямаха», их просто нет в Миассе, это ограничило многие возможности.

Но в целом первая Уральская летняя школа, несомненно, удалась, зарядив всех участников знаниями и энтузиазмом, всем хочется, чтобы УЛШИТ стала традиционной. Зависит это не только от инициативы организаторов, но и от конкретной помощи всех заинтересованных.

М. Алексеев, Н. Баталова, А. Боков  
г. Миасс Челябинской области

## Еще один объектив и чуть-чуть электроники...

...и с обычной видеокамерой «Тошиба» произойдет чудо. Теперь вместо плоской проекции окружающего мира вы сможете наблюдать на экране телевизора почти естественное изображение снятых вами сюжетов во всем их трехмерном великолепии.

Идея получения стереоизображения с помощью одновременной съемки объекта с двух точек не нова. Однако ее реализация обычно требовала двух съемочных камер и двух синхронизированных проекторов для воспроизведения изображений. Все остальное — дело оптической техники: через поляриды каждый глаз человека видит свое изображение, и с экрана прямо в зал едут поезда, скачут казаки, летят снаряды.

Прогресс видеотехники и электроники уже сегодня позволяет создавать объемное изображение с помощью всего одной видеокамеры и телевизора. Причем для воспроизведения стереовидеозаписи можно пользоваться обычным бытовым телевизором и видеоманитофоном, а для записи — стандартной магнитной лентой. Единственное устройство, которое

понадобится дополнительно — это специальные очки, разделяющие изображения для левого и правого глаза. Поставив записанную с помощью видеокамеры «Тошиба-3Д» кассету на домашний видеоманитофон, вы увидите обычную «плоскую» картинку, но стоит надеть очки, подключенные к электронному блоку управления, и экран телевизора превратится в открытое окно в мир.

## Электронный фильтр

Летать самолетами станет значительно безопаснее, если на контроле багажа в аэропортах использовать новый прибор, созданный в астрофизической исследовательской корпорации (США). Инженерами корпорации была успешно решена проблема распознавания металлических и пластмассовых изделий во время контроля ручной клади авиапассажиров с помощью рентгеновских лучей.

Аналогичные приборы, находящиеся сегодня на вооружении

таможенников, позволяют оператору увидеть на экране монитора лишь однотонное полутоновое изображение тени, образованной предметами, находящимися в сумке. И порой даже опытные контролеры не могут сразу идентифицировать оружие, замаскированное среди мелких металлических предметов, или пластиковую бомбу.

Разработанный в астрофизической корпорации прибор надежно распознает металлические предметы, пластик и другие материалы. Достичь этого позволила оригинальная система сканирования объекта сразу двумя пучками рентгеновского излучения. Встроенный компьютер перерабатывает информацию, раскрашивая полутоновое изображение в яркие цвета: металлические предметы — в голубой, органические материалы — в оранжевый, и выдает информацию на цветной монитор. Яркое изображение, которое при необходимости можно остановить и увеличить, дает значительно более полную и правдоподобную картину содержимого сумки, в которой теперь такие предметы, как пластиковые бомбы и даже снискавшие так же легко, как и металлическое огнестрельное оружие, ножи и детонаторы.

ЧТО?  
МОЖЕТ  
ЭВМ

Беляева А., Кошелюк М., Солдатова Г.  
Телекоммуникации и компьютерные  
среды

Гейн А., Линецкий Е., Сапир М., Шоло-  
хович В. Информатика: исполнители 4

## ОБЩИЕ ВОПРОСЫ

Белошапка В., Лесневский А. Основы  
информационного моделирования  
В. Г. Житомирский

Гейн А., Линецкий Е., Сапир М., Шоло-  
хович В. Информатика: алгоритмические  
конструкции 3

Гриценко В., Довгялло А. Пути развития  
информатизации образования

Глаголева Н., Юнерман Н. Основные  
понятия языка Рапира 1—4

Логико-психологические основы использо-  
вания компьютерных учебных средств  
в процессе обучения

Долматов В. Методические проблемы по-  
строения курса информатики для  
V—VI классов 6

Научная программа психолого-педагогиче-  
ских исследований по проблеме «Но-  
вые технологии обучения и развития  
детей»

Дуванов А., Зайдельман Я., Первин Ю.,  
Гольцман М. Роботландия — курс инфор-  
матики для младших школьников 5

Положение об Общесоюзном научно-ме-  
тодическом совете информатизации об-  
разования Гособразования СССР

Житомирский М., Клавдиев В. О методике  
преподавания информатики в средней  
школе 2

Решение пленума Общесоюзного научно-  
методического совета информатизации  
образования

Каймин В., Григорьев С., Угринович Н.,  
Щеголев А. Элементы логики и начала  
языка Пролог 4

Шатров А., Цевенков Ю. Проблемы  
информатизации образования

Каймин В., Кугель Л., Кузнецкий Е.,  
Точенная И. Компьютерная грамотность  
в примерах и задачах 3

Лодатко Е. Рекурсивные алгоритмы

Каймин В., Кугель Л., Кузнецкий Е.,  
Угринович Н. Алгоритмизация и начала  
программирования 5, 6

Переверзева Н., Ревчук И. Арифметические  
задачи на алгоритмическом языке

Каймин В., Угринович Н. О преподавании  
курса ОИВТ по машинному варианту 2

Полываный И. Исполнение Бейсик-  
программ

Кузнецов А., Долматов В. Методическая  
система обучения ОИВТ: структура  
и функции, состояние и перспективы 1

Савченко В. Как составить алгоритм  
Сочнев С. Путь от простого к сложному,  
или покорение алгоритма

Кушниренко А. О новом учебнике инфор-  
матики 2, 4, 6

Скобелев Г. ПМК на уроках математики  
Угринович Н. Об одном методическом  
подходе

## КВТ

Федотов В. Из рукописи конкурсного  
учебника 1, 2

Алексеев М., Алексеева Т. Доработка КУВТ  
«Агат» в школьных условиях 4

## МЕТОДИКА ОБУЧЕНИЯ

Авербух А., Гисин В., Зайдельман Я.  
Методические рекомендации к новому  
учебнику информатики

Алексеев М., Алексеева Т. Твои первые  
программы 6

Алипов Н., Соколов А. Организация  
контроля знаний

Алексеев М., Рябова А. «Агат», ассемблер  
и музыка 5

Асенова П. Изучение алгоритмической  
конструкции выбора вариантов

Аникеев В., Цвейман Н. Бейсик-букварь  
Арутюнян Г., Лашук В., Новодворский А.,  
Смирнов А., Фокин А. Система «Лектор» 2

Белов Б., Белова А. Активизация работы  
учащихся при изучении табличных вели-  
чин и массивов

Архангельский А. Мир ЭВМ 1, 4, 5

Газепов Г., Распопов В., Сульдин Ю.  
Практикум по курсу основ информатики  
и вычислительной техники на ВЦ базовых  
предприятий

Батаршев А., Рооси А. Учебный автоматизированный измерительный комплекс  
Беркуцкий В., Полываный И., Щеглов С.  
Лица «Агата» 5

Гейн А., Линецкий Е., Сапир М., Шоло-  
хович В. Информатика: как решать  
задачи с помощью ЭВМ

Вовк О. Наш кабинет 5

Гейн А., Линецкий Е., Сапир М., Шоло-  
хович В. Информатика: модели, алго-  
ритмы и исполнители

Гауенс А., Суне В. Редактор блок-схем  
Гвоздев С., Эрнстсонс Г. Использование  
плавающей арифметики Фокала в  
БК-0010 5

Гельтищева Е., Селехова Г. Гигиеническое  
обоснование сочетания цветов на видео-  
терминале 1

Городня Л. Сравнение учебных языков  
программирования Бейсик, Паскаль и Ра-  
пира 2



Гриценко А. Информационно-поисковая система для КУВТ-86  
 Гриценко А. Система подготовки текстов для КУВТ-86  
 Гришаев А. На Луну на МКШ-2  
 Гужа И. Расширение возможностей КУВТ-86  
 Гутман Г. Реализация рекурсивных алгоритмов на Бейсике  
 Дементев В. «Агат» рисует  
 Дуванов А. «Микрон»  
 Ермаков Ю. Анализ ответов: метод динамических таблиц  
 Зинюк В., Баранаскас Э. ПМК на уроке труда  
 Кадеев Ю. Язык программирования «Илюша»  
 Караваяв А. Электронный коммутатор видеосигнала для дисплейного класса  
 Карпов В., Новичков В. Графика на алфавитно-цифровом дисплее: реализация на Фортране  
 Картинка — на принтер  
 Клуб пользователей БК  
 Кузьмин Ю., Гвардина И. Система «Рига» для КУВТ-86 с Бейсиком  
 Лебедева М. Требования к содержанию и оформлению педагогического сценария  
 Манзюра В., Шмелев Д. Система для учебного моделирования языковых явлений  
 Модем для ПЭВМ — реальность  
 Монахов В. Фокал БК-0010 и его расширения  
 Новинки для системы «Рига-микро»  
 Новые возможности БК  
 Нурмухамедов Г. Знакомство с элементной базой ЭВМ  
 Очков В., Пухначев Ю. Оформление текста программы  
 Павлов С., Цилевич Б. ППС на основе математического моделирования  
 Парсаданов О., Клецо Е. Система тестового контроля знаний учащихся на БК-0010  
 Подробнее о «Микроше»  
 Работа с КУВТ-86  
 Роберт И. Средства новых информационных технологий — школе  
 Рынок ЭВМ  
 Рынок ЭВМ: Минрадиопром  
 Салимжанов Р. Стиль программирования  
 Степанова М. Компьютер и самочувствие школьника  
 Фоменко А., Гулякович М., Маланина В. Обучающая система «Страус»  
 Ханова Г., Вербицкая Л. Подключение телевизора «Рубин Ц-208» к ПЭВМ «Ямаха»  
 Шмелев А. Генератор программ-тренажеров

## ПЕДАГОГИЧЕСКИЙ ОПЫТ

Абдукадыров А., Источникова М., Вайнштейн Е. Компьютерный кроссворд в учебных целях  
 Антипов Г., Рысев Ю. Опыт использования факторного анализа на ЭВМ  
 Буловацкая М. Из практического опыта

Головская Н., Сапрыкина Г. Язык ассемблера на практических занятиях  
 Горленко В. ЭВМ и дидактические игры  
 Ерохина Р., Степанова Е. ПМК на уроках астрономии  
 Ефимова Т., Кулагина Л. Со школьной скамьи в науку  
 Земцова Л., Луканкин А. Форма или содержание?  
 Капустина Е. «Творческая лаборатория» на уроках информатики  
 Козан А. Отчего кошку назвали кошкой?  
 Крицкий А., Лецко В. Компьютерные сети в обучении  
 Кручинина Г. Нужна ли школьнику информатика?  
 Кулова Л., Мовсесян Ю. Как учить информатике сегодня?  
 Линецкий Е. Городской компьютерный центр: опыт, проблемы, перспективы  
 Раньков Е., Абдукадыров А. Решение задач с комментариями  
 Санжаров Л., Финьков А. Квазиязык на ЭВМ в учебных целях  
 Симонов А., Сысоев И., Шахова Ю. Программа-тренажер по астрономии  
 Ускова О. Школьная информатика глазами первокурсников  
 Фадеев С. Компьютер, цвет, язык  
 Штернберг Л. Уроки с программируемыми микрокалькуляторами

## ВНЕКЛАССНАЯ РАБОТА

Крылова Е., Петров М. III Московская олимпиада по информатике  
 Олимпиады по информатике  
 От игры к серьезному обучению  
 Полушкин Ю. Компьютер и азбука Морзе  
 Реунов А. Школьники выбирают дорогу  
 Шарова Н., Карасева Т. Школьники пишут программы  
 Школа в Красноярске  
 Шпилевский А. Фрактальные кластеры в модели агрегации, ограниченной диффузией

## ТОЧКА ЗРЕНИЯ

Волков Д. Школа, программирование, познание  
 Гузев В. О технологиях обучения и предмете педагогической информатики  
 Дименштейн Р., Яковлев А. Информатика или компьютерное дело?  
 Дринь Б., Скульский Р. Отбор содержания курса информатики  
 Каймин В. К концепции информатизации образования в СССР  
 Ляхович В. Практические занятия по информатике  
 Матюшкин-Герке А. О перспективах развития школьного курса информатики  
 Федотов В. Нужна ли информатика как самостоятельный предмет в общеобразовательной школе?  
 Шоломий К. О дефиците программных средств для компьютерного обучения школьным предметам

## ЭВМ В НАРОДНОМ ХОЗЯЙСТВЕ

Барсук Я., Коган Ю., Мелентьева Л.  
Компьютерное обучение в системах  
административного управления 4  
Исаева Н., Мироносецкий Н., Парфенова Л.,  
Щеглов Ю. Компьютер и хозрасчет 3

## ПЕДАГОГИЧЕСКИЕ КАДРЫ

Ефимов В. Компьютер в педвузе 6  
Зеленин В. Метод проектирования ППС  
по структурированным образцам на  
языке MSX-Бейсик 6

## ЗАРУБЕЖНЫЙ ОПЫТ

Как торгуют компьютерами в ГДР? 1  
Тихонов В. Можно ли выиграть... знания? 3  
Чан Лянтянь, Ван Чжэньшань. Компьютер-  
ное образование в школах Китая 5

## НАМ ПИШУТ

Еще раз о здоровье... 1  
Информатика и русский язык 3  
Компьютерные игры и нравственный выбор 2  
Мы верим! 3  
Нужен ли школе совместитель? 3  
О преимуществах алгоритмической нотации 2  
Открытое письмо 2  
Печальное приобретение 5  
Тема семинара — компьютеризация школы 1

## РЕПОРТАЖ НОМЕРА

Во Франции нас читают 2  
«Экос», «Электроника», «Бэй Фан»... 1

## ИНТЕРВЬЮ НОМЕРА

В творческом поиске 4  
Нужны смелые идеи, инициативные люди 4  
Я выбрал систему профтехобразования 6

## ИНФОРМАЦИЯ

Всесоюзная конференция в Севастополе 6  
Выставка, которой не могло быть три года 2  
назад 2  
Интеринформ 5  
Информатика в педучилище 5  
Два слова о съезде 3  
Конкурс «на перекрестке» 1  
Координация разработок программного  
обеспечения 2  
Конференция в Казани 1  
«Круглый стол» в Ленинграде 2  
Международная встреча в Ленинграде 3  
Офицеры обсуждают проблемы компьюте-  
ризации 5  
Семинар «Проблемы компьютеризации до-  
школьного воспитания» 5  
Семинар в Симферополе 1  
Семинар на ВДНХ 3  
Соревнуются техникумы 5  
Уральская летняя школа 6  
Школа творчества 5  
Экспертные системы в обучении 4

## ДОСТИЖЕНИЯ И ПЕРСПЕКТИВЫ

Экология мышления 6

## КНИГИ

Библиография 3  
Везет физикам 3  
Задачи по программированию 3  
ЭВМ для любителей математики 2

## БИБЛИОГРАФИЯ

По страницам тематических планов изда-  
тельств на 1990 г. 6

**МОЛОДЕЖНАЯ ИНИЦИАТИВА** 1—3, 5

**ВЕСЕЛЫЙ УРОК** 4, 5

## Венера рождается из пены

Сотрудники фирмы 3D Systems (США) разработали принципиально новый способ изготовления трехмерных деталей из пластмассы, ориентированный на использование в системах автоматического проектирования и производства.

Установка SLA-250 (Stereolithography Apparatus) включает ультрафиолетовый лазер, систему сканирования по осям X и Y,

**ЧТО?**  
**МОЖЕТ?**  
**ЭВМ**

управляющую ЭВМ. Луч лазера фокусируется на поверхности жидкого фотополимера, отверждающегося под действием ультрафиолета; ЭВМ заставляет луч обегать контуры горизонтального разреза детали. По мере формирования деталь в ванне с фотополимером погружается и луч формирует следующий слой.

Изготовление сложных деталей для аэрокосмической промышленности таким способом заняло одну неделю вместо девяти и обошлось в 4500 долл. вместо 20 000.

# ВНИМАНИЮ ГОСУДАРСТВЕННЫХ УЧРЕЖДЕНИЙ, КООПЕРАТИВОВ И ЧАСТНЫХ ЛИЦ!

Научно-производственное объединение средств вычислительной техники  
поставляет и запускает в эксплуатацию компьютерные классы  
и компьютерные центры на базе ПЭВМ «Агат» седьмой модификации («Агат-7»).

«Агат-7» имеет цветной дисплей, разрешающая способность экрана  
256×256 точек для черно-белого изображения  
и 128×128 для цветного изображения,  
текстовая информация 32×32 символа  
для 8 цветов и 64×32 для черно-белого варианта.



Основные технические характеристики ПЭВМ «Агат-7»:

разрядность шины данных — 8 бит;

разрядность шины адреса — 16 бит;

время выполнения операции типа регистр — регистр — 1 мкс;

время выполнения операции сложения — 3 мкс;

объем оперативной памяти: мин. — 32 К байт, макс. — 256 К байт;

тактовая частота системы — 1 МГц;

имеются технические средства для связи по параллельному интерфейсу,  
последовательному интерфейсу и по интерфейсу кассетного магнитофона.

У компьютера «Агат-7» есть семь разъемов

для подключения внешних устройств, что позволяет

через несколько интерфейсных плат

подключать периферийные устройства любых типов.

С помощью контроллера 121 пользователи ПЭВМ «Агат-7»

могут запускать программное обеспечение для «Эплл-II».

«Агат-7» имеет накопители на гибких магнитных дисках

емкостью 140 К байт и 1000 К байт,

2 выносных пульта (джойстика),

можно подключать плату контроллера для 800 К-байтного

двухстороннего дисководов,

есть возможность записи на магнитную ленту, и ввод с телекамеры.

Объединение заключает договоры на разработку и поставку

на базе ПЭВМ «Агат-7» автоматизированных рабочих мест

типа «Руководитель», «Кадры», «Поликлиника», «Библиотека» и другие,

комплектуя их самым современным периферийным оборудованием:

принтерами, накопителями типа «Винчестер».

**Стоимость ПЭВМ «Агат-7» 3900 рублей.**

НПО СВТ заключает договоры на гарантийное

и послегарантийное обслуживание и ремонт —

это решит все Ваши проблемы.

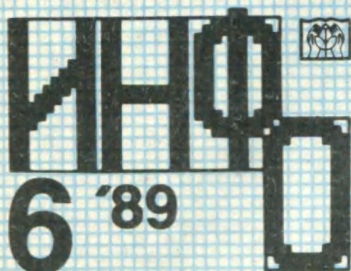
**Наш адрес: 117485, г. Москва, ул. Волгина, д. 33.**

**тел. 930-68-77.**

60 коп.  
Индекс 70423

3/4

OldPC.ru  
7002  
музей компьютеров



ИНФОРМАТИКА  
И ОБРАЗОВАНИЕ

