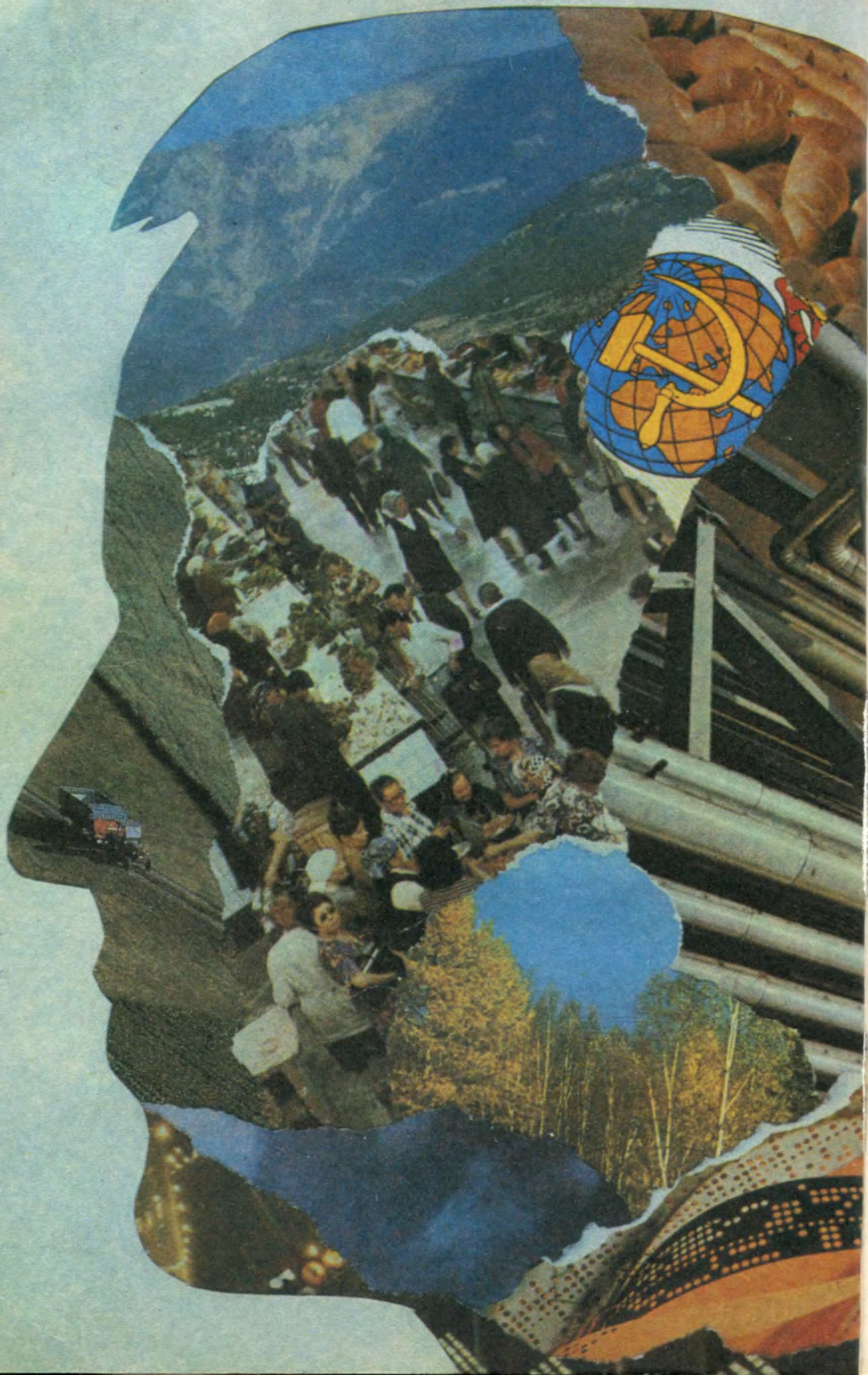


# ИНФОРМАТИКА И ОБРАЗОВАНИЕ

1990







# ИНФОРМАТИКА И ОБРАЗОВАНИЕ

1'90  
январь —  
февраль

OldPC.su  
7 0 0 3  
музей компьютеров

## Содержание

### Общие вопросы

Концепция информатизации образования	3
Перегудов Ф. Системная деятельность и образование	10
Бургин М., Степенко Г. Информационный поиск и компьютерная грамотность	15

### Методика обучения

Гейн А., Линецкий Е., Сапир М., Шолохович В. Информатика: циклы и вспомогательные алгоритмы	23
Гольцман М., Дуванов А., Зайдельман Я., Первин Ю. Информация вокруг нас	29
Ляхович В. Методика составления алгоритмов	39

### Кабинет ВТ

Решение задач с помощью Бейсика	48
Борисов В., Поляков С. Использование языков высокого уровня на КУВТ-86	55
Коцоба Е. Сделайте АОС сами!	57
Боксер О., Васильченко А., Прияткин А. Реакциометрические обследования на учебных ПЭВМ	59
Файловый монитор FILMON для ПЭВМ «Микроша»	60
Коробицын Д. Изучаем Бейсик за полчаса. Новые пособия	61
Клуб БК	65

### Педагогический опыт

Штернберг Л. Уроки с программируемыми микрокалькуляторами	75
Фадеев С. Компьютерные игры и орфография	81
Меш Г. ПЭВМ в помощь учителю химии	84
Лазарева Е. Помогает центр информатики	86

Энциклопедия

САНКТ-ПЕТЕРБУРГСКОГО ПЕДАГОГИЧЕСКОГО УНИВЕРСИТЕТА

## Внеклассная работа

- Кузнецов С., Распопов В. Формирование навыков параметрического программирования 87  
Зайцев А. Спутники, компьютеры, образование 91  
Ландо С., Манакова О. Компьютер в летнем лагере 94

## Молодежная инициатива 99

## Зарубежный опыт

- Нокс Дж. Что могут дать компьютеры педагогике: Взгляд из американской школы 107

## Нам пишут

- Сердитое письмо 113  
Личность учителя 114  
Доступно уже сегодня 115  
Вам интересно? 116  
Почтовый ящик 117

## Информация

- Второй пленум Общесоюзного научно-методического совета информатизации образования 120  
Лебедева М. Международное совещание в Гатчине 122  
Болгаро-советский семинар 123  
Ждем продолжения 124

## Книги

- Алгоритмический язык — примеры и задачи 127  
Через задачи — к программированию 127

## Веселый урок

- Жиров А. Общедоступный словарь по вычислительной технике 128

Главный редактор  
академик  
**В. А. МЕЛЬНИКОВ**  
Редакционная  
коллегия  
**И. Н. АНТИПОВ**  
**В. Н. АФАНАСЬЕВ**  
**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)

Обложка Э. Бажиллина

Редактор отдела А. Кравцова  
Научный редактор Т. Драгныш  
Зав. редакцией Н. Игнатова  
Художественный редактор Л. Розанова  
Корректоры Е. Власова, О. Пурлова

Сдано в набор 21.11.89. Подписано в печать 04.01.90. А 03697  
Формат 70×100 1/16. Бумага офсетная № 1. Печать офсетная. Усл. печ. л. 10,40.  
Усл.-кр. отт. 42,88. Уч.-изд. л. 13,50. Тираж 71 180 экз.  
Заказ 2770. 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., 19, кор. 1, кв. 17.  
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», 1990

## Концепция информатизации образования

Концепция информатизации образования (далее — концепция) формулирует отправные положения начинающегося процесса перестройки образования в условиях информатизации общества, определяет цели, возникающие задачи и намечает основные направления их решения с указанием этапов, промежуточных целей и требуемых ресурсов. Изложенные в концепции положения относятся ко всем уровням и учреждениям системы непрерывного образования в нашей стране. Концепция информатизации образования является составной частью концепции информатизации общества.

### 1. Информатизация общества и образования

Информатизация общества — объективный процесс, связанный с повышением роли и степени воздействия интеллектуальных видов деятельности на все стороны жизни человечества. Отличительной чертой современного периода мирового развития является перемещение центра тяжести в общественном разделении труда из сферы материального производства в область получения, переработки, передачи, хранения, представления и использования информации. Информатизация становится таким же страте-

гическим ресурсом общества и государства, как традиционные материальные или энергетические его ресурсы. Формируются новые информационные технологии (НИТ), вводящие в обиход методы сбора, хранения, обработки, передачи и представления информации с использованием компьютеров.

Информатизация, в широком смысле слова, представляет собой процесс перестройки жизни общества на основе все более полного использования достоверного, исчерпывающего и своевременного знания во всех общественно значимых видах человеческой деятельности. Этот процесс предполагает формирование новой инструментальной базы человеческой деятельности — инфраструктуры средств накопления, хранения, обработки и передачи информации. Информационная оснащенность, масштабы и эффективность использования средств связи и информатизации уже вошли в число важнейших показателей уровня научно-технического прогресса нашей страны.

Информатизация образования является ключевым условием успешного развития процессов информатизации общества и требует приоритетного обеспечения ресурсами.

Информатизация образования — процесс подготовки человека к полноценной жизни в условиях информационного общества.

Проникновение НИТ в сферу образования позволит педагогам качественно

Одобрена решением второго пленума Общественного научно-методического совета информатизации образования

изменить содержание, методы и организационные формы обучения. Информатизация образования является не только следствием, но и стимулом развития НИТ, способствует ускоренному социально-экономическому развитию общества в целом. Новые информационные технологии в образовании способствуют:

раскрытию, сохранению и развитию индивидуальных способностей обучаемых, присущего каждому человеку уникального сочетания личностных качеств; формированию у учащихся познавательных способностей, стремления к самосовершенствованию;

4      обеспечению комплексности изучения явлений действительности, неразрывности взаимосвязи между естествознанием, техникой, гуманитарными науками и искусством;

постоянному динамичному обновлению содержания, форм и методов процессов обучения и воспитания.

## 2. Основные этапы процесса информатизации образования

Информатизация образования — длительный и сложный процесс. Он связан с созданием опережающего научно-методического задела, переподготовкой работающих и подготовкой новых поколений педагогов, развитием необходимой материально-технической базы, становлением новой культуры педагогического труда. Ход этого процесса при возможных существенных различиях, связанных с территориальными особенностями и неравномерностью развития разных учебных заведений, характеризуется тремя последовательными этапами. На первом:

начинается массовое освоение средств новых информационных технологий и прежде всего компьютеров;

разворачивается исследовательская работа по их педагогическому освоению, развитию ранее найденных, но неприменимых при традиционной технологии обучения методов и организационных форм учебной работы, поиск новых составляющих содержания образования;

происходит осознание обществом сути объективной необходимости процессов информатизации.

Одной из основных точек кристаллизации этого процесса сегодня является базовая подготовка в области информатики на всех ступенях системы непрерывного образования.

По мере развития в стране технической базы поддержания инфосферы и распространения соответствующих технических средств (компьютерных лабораторий, средств телекоммуникаций, оперативной полиграфии, систем интерактивного видео, баз данных и т. п.) в учебных заведениях начинается следующий этап этого процесса. Его основное содержание:

активное освоение и фрагментарное внедрение средств НИТ в традиционные учебные дисциплины и на этой основе — массовое освоение педагогами новых методов и организационных форм учебной работы;

практическая постановка вопроса о радикальном пересмотре содержания образования, традиционных форм и методов учебно-воспитательной работы;

разработка и начало освоения систем учебно-методического обеспечения («программно-методических комплексов», «компьютерных курсов»), включающих программные средства для ЭВМ, различные видео- и аудиоматериалы, тексты для учащихся и методические материалы для педагогов. При изучении «компьютерных курсов» традиционные организационные формы и методы обучения свободно сочетаются с индивидуальными, групповыми и лекционными формами работы, с занятиями в компьютерных, физических и других лабораториях.

Характерной особенностью второго этапа станет повсеместное использование средств НИТ, поддерживающих все многообразие возможных форм организации учебно-воспитательного процесса.

На третьем этапе произойдет радикальная перестройка содержания непрерывного образования на всех его ступенях, обусловленная процессом информатизации общества, смена методической основы обучения, освоение каждым педагогом широкого спектра конкурирующих и взаимодополняющих методов и организационных форм обучения, поддерживаемых соответствующими средствами НИТ.

Смена этапов — неодномоментный процесс. Различные регионы и учебные заведения будут проходить его в разное время. Главное — поддержать его непрерывность, стимулировать тех, кто находится на переднем крае работы, создавая на практике систему образования завтрашнего дня.

### 3. Информатизация и изменение содержания обучения

Изменение целей и содержания обучения является ведущим звеном процесса информатизации образования. Технологическое переоснащение учебного процесса, появление новых (а чаще всего выход в широкую практику давно найденных, но не нашедших распространения по экономическим, технологическим и другим соображениям) методов и организационных форм обучения является производным, обеспечивающим достижение выдвигаемых целей.

Обновление содержания обучения связано прежде всего с повышением роли гуманитарной подготовки, формированием у учащихся последовательного естественнонаучного представления об окружающем мире.

Изменение содержания обучения идет по нескольким направлениям, значимость которых меняется по мере развития процесса информатизации общества.

Первое направление связано со становлением учебных дисциплин, обеспечивающих общеобразовательную и профессиональную подготовку учащихся в области информатики.

Второе — с расширяющимся использованием средств информатизации, применение которых становится нормой во всех областях человеческой деятельности. Этот процесс влечет за собой изменение предметного содержания всех учебных дисциплин на всех уровнях образования.

Третье направление связано с глубинным влиянием информатизации на цели обучения. Оно будет все более ощущаться по мере развития процессов информатизации общества, проведения работы по реструктурированию накопли-

ваемых человечеством знаний, формирования в общественном сознании представлений об энциклопедической природе образования, необходимого каждому гражданину. Предстоит выработать качественно новую модель подготовки членов будущего «информационного общества», для которых способность к человеческим коммуникациям, активное овладение научной картиной мира, гибкое изменение своих функций в труде, ответственная гражданская позиция, творческое мышление и развитое планетарное сознание станут очевидной жизненной необходимостью.

Развертывание работ по каждому из выделенных направлений определяется достигнутым уровнем развития общества и связано с этапами его информатизации.

### 4. Развитие методов и организационных форм обучения в условиях информатизации образования

Информатизация образования создает предпосылки для широкого внедрения в педагогическую практику психолого-педагогических разработок, позволяющих интенсифицировать учебный процесс, реализовать идеи развивающего обучения, в том числе в рамках новых моделей. Развитие методов и организационных форм обучения обусловлено возможностями НИТ как инструмента человеческой деятельности и принципиально нового средства обучения.

НИТ открывают каждому обучаемому доступ к практически неограниченному объему информации и ее аналитической обработке, что обеспечивает «непосредственную включенность» в информационные потоки общества. НИТ представляют собой универсальное средство познавательно-исследовательской деятельности, являются вторым по значимости, после традиционной письменности, знаковым орудием, обеспечивающим оперативный обмен информацией по содержанию выполняемой деятельности.

Педагогически целесообразное использование НИТ позволяет усиливать интеллектуальные возможности учащегося, воздействуя на его память, эмо-

ции, мотивы, интересы, создает условия для перестройки структуры его познавательной и производительной деятельности.

При этом изменяется роль педагога, основная задача которого — поддерживать и направлять развитие личности учащихся, их творческий поиск, организовывать их совместную работу.

Радикальные изменения в методику учебной работы внесет распространение спутниковой связи, систем интерактивного видео, локальных баз данных большого объема на компакт-дисках и т. п.

6 Распространение НИТ в сфере образования — мощный рычаг формирования личностных качеств учащихся, необходимых для жизни в условиях, порождаемых информатизацией общества. Решающее значение здесь имеет система социальной организации учебно-воспитательного процесса, которая переводит требования технологии на язык культуры взаимоотношений между людьми. Естественной культурной средой НИТ в сфере образования является педагогика сотрудничества. Ее сочетание с НИТ обеспечивает углубление и развитие человеческих контактов между всеми участниками учебно-воспитательного процесса, создает наиболее благоприятные условия для всестороннего и гармонического развития личности.

В этих условиях неизбежен пересмотр сложившихся сегодня организационных форм учебной работы: увеличение самостоятельной, индивидуальной и групповой работы учащихся, отход от традиционного урока с преобладанием объяснительно-иллюстративного метода обучения, увеличение объема практических и лабораторных работ поискового и исследовательского характера, внеаудиторных занятий, которые будут обязательной составной частью целостного учебного процесса.

Важным условием эффективного использования НИТ является правильное исходное представление у педагогического коллектива о месте и роли НИТ, которое не должно быть ограничено применением их только на уроках информатики, а привести к повсеместному использованию этих средств в учебно-воспитательном процессе.

## 5. Информатизация и подготовка педагогических кадров

Информатизация образования предъявляет новые требования к профессиональным качествам и уровню подготовки педагогов, требует существенной перестройки в их работе, значительных усилий для повышения самостоятельности учащихся, создания на занятиях атмосферы творческого поиска и делового сотрудничества.

Педагогическое освоение НИТ включает:

овладение основами необходимых знаний и накопление личного опыта их практического использования;

общекультурную и методическую подготовку по их использованию в учебном процессе.

Наибольшую практическую ценность сегодня имеет приобретение педагогических навыков эффективного использования компьютеров и других средств информатизации, создания электронных библиотек, ведения справочников и архивов. Задача обеспечения компьютерной грамотности носит подчиненный характер по отношению к задаче формирования информационной культуры и методической подготовки педагогов для использования НИТ в учебном процессе.

Подготовка, переподготовка и повышение квалификации педагогических кадров в области использования НИТ в учебно-воспитательном процессе должны проводиться в исследовательских центрах практико-ориентированной педагогической науки, на кафедрах и в учебных заведениях, ведущих интенсивную экспериментальную работу в области педагогического использования НИТ.

Переподготовка педагогических кадров должна стать основной формой внедрения результатов педагогических исследований в практику для всех исследовательских групп. Предоставив каждому учебному заведению широкий выбор мест, где могут повысить свою квалификацию педагоги, и возложив на него экономическую ответственность за сде-



ланный выбор, можно создать разветвленный рынок конкурирующих между собой педагогических технологий, эффективно поддерживать наиболее продуктивные исследовательские проекты.

Значительное внимание следует уделить подготовке и переподготовке в области НИТ педагогических кадров высшей квалификации. Здесь для проведения занятий наряду с лучшими отечественными специалистами необходимо привлекать и зарубежных педагогов, имеющих опыт использования НИТ в соответствующей предметной области.

Существенную помощь в решении задач подготовки, переподготовки и повышения квалификации педагогических кадров может оказать использование НИТ. Во все программные средства, используемые в сфере образования, необходимо включать специальное педагогическое обеспечение, позволяющее каждому педагогу быстро и эффективно осваивать их. Новейшие разработки (системы интерактивного видео, базы данных на компакт-дисках и т. п.) должны в первую очередь использоваться для подготовки и переподготовки педагогов.

Развитие средств коммуникации (телеконференции, электронная почта и т. п.), в том числе на основе спутниковой связи, дает возможность педагогам контактировать со своими коллегами в любой части света. Это открывает широчайший простор для создания коллективов педагогов-единомышленников. Сочетание средств «обучения на расстоянии» с периодическими личными контактами обеспечит проникновение новых педагогических методов и идей в каждое учебное заведение.

Подготовка и переподготовка педагогических кадров являются приоритетным направлением решения задач информатизации образования и должны обеспечиваться всеми необходимыми ресурсами в первую очередь. Широкое использование НИТ в педагогических вузах и университетах, центрах переподготовки и повышения квалификации педагогов является одним из условий повышения общественного статуса педагогического труда, одним из условий успеха процесса информатизации образования.

## 6. Информатизация и перестройка управления образованием

Радикальное изменение сложившейся практики управления в системе народного образования является решающим условием успеха происходящих в образовании реформ. Средства НИТ дают возможность массовой реализации уже наработанных в педагогике концепций, которые не были осуществлены из-за отсутствия необходимого инструментария. Облегчая и ускоряя процессы переработки информации, упрощая взаимодействие между людьми, НИТ становятся сегодня мощным средством перестройки управления образованием, а сама перестройка — необходимым условием их эффективного использования.

Сформировавшаяся установка на гибкость и многовариантность учебно-воспитательного процесса, управление учебными заведениями со всесторонним учетом их специфики и местных особенностей позволяет отказаться от бытующего еще стиля внедрения компьютеров, при котором единообразные решения тиражируются без учета реалий, существующих на местах.

Процесс информатизации в различных регионах страны будет развиваться неравномерно. Новые информационные технологии приживаются прежде всего там, где имеется развитая сеть коммуникаций, налажена сфера информационного обслуживания, действует служба технического сервиса, имеются компьютерные клубы, научно-технические общества и другие объединения, где пользователи новой техники могут обмениваться опытом, вырабатывать коллегиальные решения. Здесь возникает разветвленная сеть «точек роста», где педагоги-новаторы ищут приемы повышения эффективности учебно-воспитательной работы с использованием НИТ. Их многообразие позволяет каждому учебному заведению выбрать модель развития, в наибольшей степени отвечающую его конкретным условиям. Формирующиеся вокруг «точек роста» творческие коллективы должны соревноваться между собой за выделение целевого финансирования на проведение исследовательских работ, в первую очередь получать

новые образцы технических средств, служить базой для переподготовки и повышения квалификации педагогических кадров. Этот процесс должен направляться и координироваться местными органами управления и стимулироваться соответствующими решениями регионального уровня.

Поддержка «точек роста», развитие исследовательских проектов, создание моделей учебных заведений завтрашнего дня — главная задача централизованных программ. Вместе с тем эти программы должны предоставлять возможность участия в процессах информатизации каждому педагогу, который хочет и может внести свой вклад. Обеспечение теледоступа к удаленным базам данных, создание систем спутниковой связи делают решение этой задачи реальным уже сегодня. Другая задача централизованных программ — проведение единой государственной политики в области оснащения учебных заведений средствами информатизации и выработка экономических механизмов для их эффективного использования.

Основная цель местных программ — обеспечить повышение уровня работы конкретных учебных заведений, стимулируя использование НИТ там, где это приносит наибольший педагогический эффект. Главные задачи здесь:

выбрать модель внедрения НИТ, в полной мере отвечающую местным условиям;

организовать подготовку и переподготовку педагогических кадров;

оснастить учебные заведения необходимыми методическими разработками, техническими и программными средствами, обеспечить их эксплуатацию.

Качественно новым в управлении образованием может стать широкое распространение программно-целевого метода планирования и управления. Его внедрение позволит работникам управления всех уровней последовательно и систематически оценивать текущие и предлагаемые проекты и мероприятия с точки зрения их вклада в достижение стоящих перед организацией целей, распределять наличные ресурсы в полном соответствии с приоритетами имеющихся задач. Использование программно-целевого

планирования позволяет избежать потерь, вызываемых тем, что расчет потребных и распределение имеющихся ресурсов производится в отрыве от определения целей и задач, стоящих перед педагогическими коллективами. Программно-целевое управление позволит правильно оценивать сравнительную эффективность различных мероприятий, наиболее продуктивно использовать имеющиеся ресурсы.

## 7. О ресурсном обеспечении процессов информатизации образования

Информатизация образования требует значительных бюджетных средств для подготовки и переподготовки кадров; приобретения, установки и обслуживания средств информатизации; разработки производства и поставки программных продуктов.

Следует сочетать традиционную практику использования централизованных государственных вложений с широким привлечением других источников финансирования: средств местных Советов, предприятий и т. п.

Существующая в настоящее время практика оснащения системы народного образования программно и аппаратно несовместимыми типами компьютеров приводит к разобщению образовательных учреждений не только по вертикали, но и по горизонтали, нерациональному использованию выделенных средств, снижает эффективность использования техники, а низкая эксплуатационная надежность технических средств дискредитирует саму идею информатизации образования.

Суммарные затраты на информатизацию образования в ближайшее десятилетие составят миллиарды рублей. Для их эффективного использования доля затрат на проведение всех видов научно-исследовательских работ в общем объеме ассигнований должна существенно возрасти и составить 5—7 % выделяемых ассигнований. До трети выделяемых средств необходимо расходовать на переподготовку и повышение квалификации педагогических кадров. Затраты на приобретение различных средств информатизации и соответствующих учебно-

методических материалов являются основными лишь на первом этапе процесса.

Рациональная структура затрат должна быть тщательно выверена и обоснована в соответствующих государственных и региональных программах. Держателем этих средств, заказчиком средств информатизации для системы образования должен стать соответствующий государственный орган, республиканские и местные органы управления образованием.

Основная доля средств на приобретение техники и ее эксплуатацию, на программные средства и переподготовку кадров должна распределяться через местные Советы. Дополнительное финансирование программ и проектов из централизованных источников должно стать эффективным средством проведения государственной политики в области информатизации образования. Централизованное финансирование должно обеспечивать выполнение комплексных и широкомасштабных проектов в интересах системы образования в целом. Эти работы должны вестись на конкурсной основе, с широким привлечением всех заинтересованных организаций, в том числе совместных предприятий, кооперативов и инициативных групп, иностранных фирм.

Процесс информатизации образования в нашей стране только начинается. Сегодня НИТ распространяются в науке, в производстве, в военном деле. Завтра они начнут занимать свое место в сфере финансов, в торговле, в быту. Использование их в учебном процессе должно поддерживать позитивные сдвиги во всех звеньях системы непрерывного образования.

Развитие НИТ идет как никогда быстро. Объем памяти массовых персональных компьютеров в 80-е гг. возрос в десятки раз. Качественно изменяются характеристики внешних запоминающих устройств, емкость и надежность которых возрастают в сотни раз. Впереди — объединение компьютеров в видеосистемами. Все более доступными для использования в обучении становятся средства телекоммуникации.

Реализация программы информатизации образования поставила задачу создания развитой инфраструктуры связи, в первую очередь спутниковой, обслуживающей учебные заведения всех типов на территории страны.

На современных персональных компьютерах появилось новое поколение программных средств, которое в большей степени отвечает познавательным способностям людей, обеспечивая произвольное совместное использование текстовой, графической, звуковой информации и движущихся изображений.

Все это создает базу для качественного преобразования методов учебной работы, помогает на практике изменить содержание обучения в соответствии с новыми целями, стоящими перед нашим изменяющимся обществом. В какой мере удастся реализовать этот потенциал, зависит от принимаемых сегодня решений.

*Проект подготовили:*

*Алгинин Б. Е., Киселев Б. Г., Ландо С. К., Орешков И. С., Рубцов В. В., Семянинов Б. Г., Уваров А. Ю., Черешкин Д. С., Айламазян А. К., Хорошилов В. О., Голяс Ю. Е., Коптелинин В. Е., Новоселова С. Л., Роберт И. В., Рыбаков И. М., Трусов Б. Г., Цевенков Ю. М., Христочевский С. А., Коздов Н. М., Завальский Ю. С.*

## Системная деятельность и образование

В новых условиях человеческого общежития складывается парадоксальная ситуация, когда, располагая определенными знаниями, мы оказываемся не способными эффективно их использовать, обращаться к знаниям, как капитал, в дело. Все больше наблюдается малообоснованных действий образованных людей, не исключая тех, кто принимает ответственнейшие решения. Не выручает и такая понятная и простая на первый взгляд каноническая схема мышления: идти от частного к целому, учитывать всеобщую взаимосвязь явлений.

Любое решение хорошо ко времени, и чем изменчивее обстановка, тем времени для выбора меньше. Дефицит времени провоцирует дополнительные ошибки, оборачивающиеся серьезной бедой, если речь идет о решениях долгосрочных. Не выручает и спасительная кое для кого мысль о том, что бездействие порой лучше необдуманного действия, мол, лучше вообще не принимать никаких решений, — на самом деле промедление на руку отрицательным тенденциям.

Традиционно считается, что от неудач человека всегда защищали опыт (свой или чужой), а также наличие здравого смысла. Это должно получиться, потому что так уже не раз получалось у меня или у других, — подбадривают себя одни. Другие, напротив, предостерегают: этого не должно быть потому, что так никогда не было, так никто никогда не делал, и т. п. Хотя порой пренебрегают и здравым смыслом. Иначе чем объяснить усиливающуюся гонку вооружений, взаимотчужденность людей вплоть до национальной розни, неспособность обдумать хотя бы два шага вперед, когда речь идет, к примеру, об управлении экономикой — пагубность шарахания от полной централизации к полной децентра-

лизации мы уже испытываем... Возможно, в мало меняющейся среде для решений, все последствия которых ограничивает время, подобные рассуждения справедливы. Знание прошлого и взаимосвязь явлений в этих случаях легко (линейно) экстраполируются из прошлого на будущее, надежно срабатывают рецептурные справочники, советы бывалых людей, рекомендации менеджеров.

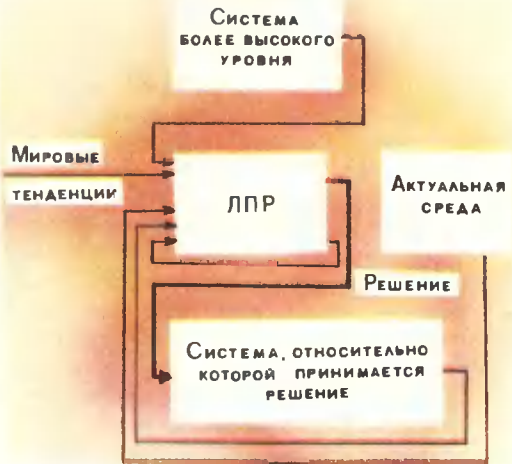
Однако современный мир нестабилен. Коммуникации, информационные технологии резко изменяют его вид, количество и характер многовековых связей между людьми и социальными группами. Грубое вмешательство в природную среду разрушает казавшиеся незыблемыми связи в окружающем нас мире. Прimitивная экстраполяция опыта прошлого зачастую не срабатывает и порождает новые ошибки; не выручает и здравый смысл.

На наш взгляд, все эти проблемы носят глубинный характер и объясняются кризисом современного образования. Информационный бум, сопровождаемый большим объемом избыточной информации, или, как говорят специалисты, зашумленностью общего фона, усугубляет и без того порочную тенденцию: в ограниченное время обучения втискивать все больше разноплановых, лоскутных сведений вместо того, чтобы фиксировать и укреплять предметные и особенно межпредметные связи. Незнание и непонимание причинно-следственных связей, отсутствие у педагогов и обучаемых навыков и даже интереса к формированию подвижных проблемно-ориентированных, интегрированных баз знаний «под задачу» не позволяют знания превращать в умения. Выход видится нам в ревизии целей и содержания современного образования на основе принципов системной деятельности. Заодно такой подход поможет воссоединить многочисленные, но пока разрозненные примеры новаторства в педагогике, т. е. бессистемность сама по себе неэффе-

тивна, а здесь она порождает новые противоречия и беспочвенные дискуссии типа «чей опыт лучше». Деятельность каждого образованного человека должна быть системной, и уже по одной этой причине обучение ей должно быть важной составной частью современного образования.

Системная деятельность подразумевает использование совокупности взаимосвязанных логических (алгоритмических) и аналитических процедур. Они облегчают человеку анализ реальных жизненных ситуаций, предваряющих разумный выбор цели своей деятельности, способов ее достижения и организационных форм реализации, позволяют объективно сопоставлять желаемое с фактически достигнутым. На любом этапе системная деятельность, хотим мы того или не хотим, навязывает нам процедуру выбора того или иного решения. Из опыта известно, что наиболее уязвимое звено наших действий — выбор цели, поскольку ошибка в выборе цели сводит на нет всю остальную работу.

Итак, чтобы решать — надо выбирать; чтобы выбирать — надо перебирать варианты; чтобы перебирать варианты — надо моделировать. Выбор — один из основных актов умственно-логической деятельности человека. В этот момент им движут личные стимулы, интересы и ценностные ориентиры. Он, как умеет, пытается соотносить свои желания и свои ресурсы. Кто-то ведет себя при этом трезвым прагматиком, а кто-то остается романтиком и фантазером. Эгоист в процессе выбора стремится установить «нулевые веса» против чужих интересов. И этот крайний случай, пусть даже на практике не так частый, важен, поскольку выражает позицию экстремизма. Однако в большинстве случаев лицо, принимающее решение (ЛПР), идет на компромисс и помимо собственных интересов учитывает еще и интересы тех, на ком отразится будущее решение. Здравомыслящий человек считается и с общественным мнением, и с установившимися на сей счет традициями (рис. 1).



нительба, проведение боевой операции или выбор новой продукции для предприятия). Успех или неуспех равно не проходит для нас без следа. Говорят, чтобы слыть удачником, надо принимать хотя бы 51 % правильных решений. В случае неудачи мы либо не достигаем поставленной цели вообще, либо проигрываем во времени, либо имеем дело с неожиданными последствиями, перечеркивающими результаты ранее проделанной работы.

Системная деятельность по самому своему определению есть эффективный и надежный метод организации знаний о реальных объектах и их поведении, своеобразный способ для обращения знаний в умение жить (в отличие от расхожего смысла). Она — залог высокого качества интеллектуальной жизни личности.

Вкус к системной деятельности в нашем сложном реальном мире следовало бы прививать с малых лет. Разделение на простое и сложное достаточно условно. Строго говоря, понятия о сложных и простых системах существуют применительно в отношении к моделям, объясняющим и прогнозирующим их поведение. Если модель позволяет с достаточной точностью предсказывать будущее системы, она уже полезна, а система проста, хороша и понятна. Задача системной деятельности в том и состоит, чтобы превращать сложное в простое

Проблема выбора особенно остра в нестандартной ситуации, какой бы ни разыгрывался сюжет (смена работы, же-

(но не наоборот) и тем облегчать нам жизнь. Кстати, это ее роднит с настоящей наукой. Искусство системного анализа состоит в том, чтобы создавать минимально избыточные, но адекватные модели систем, ранее считавшихся сложными, которые позволяют проиграть сложную жизненную ситуацию и найти оптимальный выход. После чего выявленные закономерности становятся обыденными, используются уже без ссылки на то, что это делается системно. Многократно проверенные правила, рецепты, приемы выбора и составляют в конечном счете традиционный опыт, передаваемый из поколения в поколение. Он закладывается в образовательные системы, но, к сожалению, без объяснения причин и источников своего происхождения.

Все ранее сказанное приводит к мысли, что необходимо иначе взглянуть на образование. Оно должно стать средством для достижения безопасного и комфортного существования личности в современном динамичном мире, способствовать гармонии ее отношений с окружающей средой. Но для этого нужно осознанное поведение. Очевидно, что динамичный мир нельзя адекватно отразить застывшей образовательной системой — образование должно стать непрерывным. Поскольку любая личность неповторима, то в идеале каждый человек имеет право на дружественную ему систему образования, продвижение по «собственной образовательной траектории», которой не чужды индивидуальный подход, постоянная психодиагностическая и профориентационная поддержка.

Высший смысл пребывания человека в образовательной системе — максимальное раскрытие способностей и воспитанность, проявляющиеся в его реальном поведении. Без перехода от обязательного обучения на первоначальных этапах к умению учиться самостоятельно в течение всей последующей жизни сделать это не удастся. Переход от обучения к научению, приобретению умений и навыков (неважно — чему учили, важно — чему научили) является принципиальным.

Научение эффективному, обоснованному выбору предполагает своевремен-

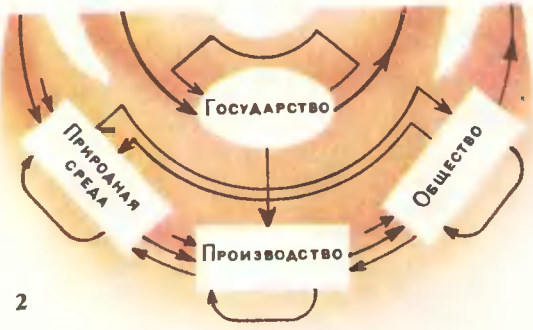
ное и качественное обеспечение этой процедуры: от конкретных базовых (обязательных, фундаментальных) знаний к познанию структур баз знаний, распределенных на многих информационных носителях (знать, что нужно, кто это знает, где это есть).

Заметим, что с расширением возможностей компьютеров мы все чаще задаем себе вопрос: как распределить знания между памятью компьютера и своей собственной?

Итог обязательной для каждого обучаемого фундаментальной подготовки — познание основных закономерностей явлений живой и неживой природы, осмысленных и трактуемых на уровне их связей между собой. Должен работать принцип «черного ящика», объясняющий пользователю, что будет на выходе, если на вход воздействовать каким-то образом... Такие представления должны развиваться последовательно, от целого к частному, начиная с глобальных моделей социально-экономических и экологических систем (рис. 2). Само место обучаемого, поставленного в центре модели, помогает наглядно ощутить глобальный, всепроникающий характер связей (рис. 3). Из их перечисления и классификации и должен появиться перечень необходимых обязательных дисциплин (физика, биология, история и т. д.). Осознание иерархической совокупности взаимосвязанных моделей (основные законы), позволяющее воспринимать окружающий мир во всем единстве и многообразии, — это и есть то, что принято считать мировоззрением. Без него фундаментальная подготовка невозможна. Речь идет о первом уровне познания, ответственном за безопасность человека в окружающем мире (рис. 4).

На втором уровне, который учащийся волен выбирать сам, определяя будущую профессиональную деятельность, необходимо более детальное познание отдельных объектов (систем) для их грамотного использования (эксплуатации) и простейшего ремонта. Комплексный характер познания объектов подразумевает изучение разных типов закономерностей (физических, биологических и т. п.).

Третий уровень познания (анализ и

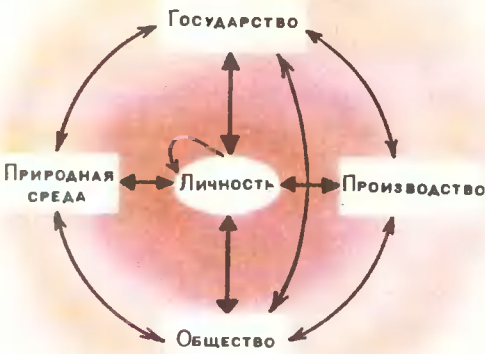


2

тически вся подобная деятельность носит коллективный характер, следовательно, умение искать и находить разумный компромисс при объединении усилий людей в достижении общей задачи — ценнейшее качество личности. Однако учебный процесс повсеместно игнорирует эту тривиальную истину. Учащихся творческой, совместной работе практически не учат.

Известно также, что процесс познания идет тем успешнее, чем скорее человеку отводится роль не объекта, а субъекта обучения. Напомним, что учебный процесс (рис. 5) может быть представлен объемной моделью, включающей в себя обязательными элементами преподавателя, учащегося, изучаемый предмет и средства обучения. Четыре плоскости этой пирамиды отражают многообразие учебного процесса: аудиторную и самостоятельную работу учащегося, консультационную — под руководством преподавателя и, наконец, регулярную методическую работу самого учителя. Из них пока в почете — первая, самая традиционная, уходящая корнями в далекое прошлое, когда все знания учащийся получал только от преподавателя. В этой односторонней «плоской» модели учащийся всегда только объект обучения, по этой причине он никогда не получит

13



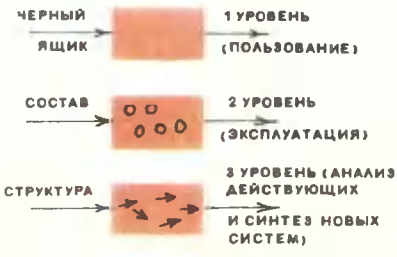
3

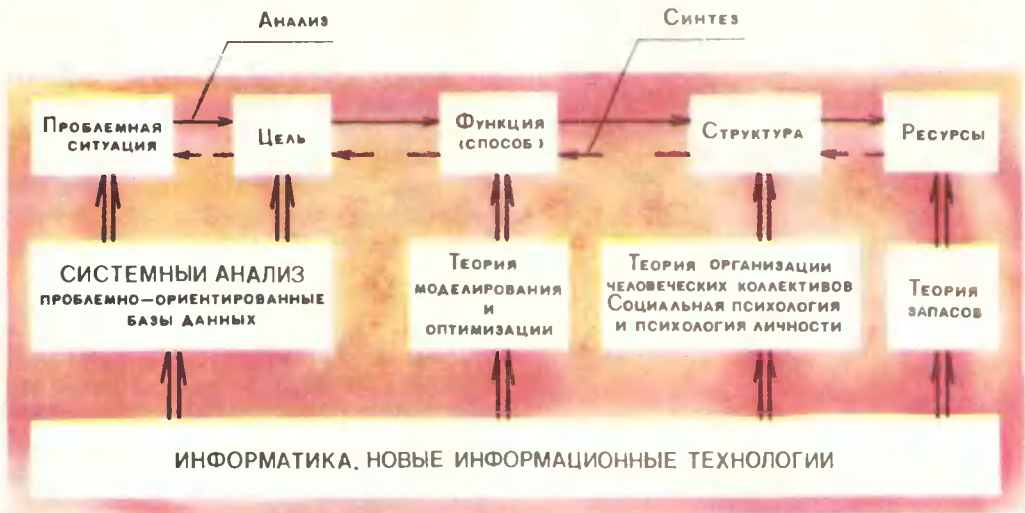
синтез систем) ориентирован на творчество и созидательную деятельность. Структурно-функциональный анализ существующих систем и синтез новых — главная задача познания этого наиболее сложного уровня. Здесь особенно важны коллективно-деятельностный характер обучения и его проблемная направленность. Нельзя в этой связи не обратить внимание на открывшееся противоречие между наиболее распространенными методами организации учебного процесса и реалиями последующей деятельности человека в обществе. Прак-

5



4





14 вкуса самостоятельного познания, не научится учиться.

Проблемная направленность обучения нужна для того, чтобы научить выбору, принятию решений. В этой процедуре полезен алгоритм модели выработки решений (рис. 6). Его главный смысл — идти при анализе от проблемы к ресурсам через последовательный выбор цели, эффективного способа ее реализации, организационной структуры. Постоянно соизмерять желания и наличные ресурсы. При синтезе вновь создаваемой или реконструируемой системы проходить обратный путь — от выделенных ресурсов до ликвидации проблемы. Для успешной реализации такой деятельности, как следует из приведенной схемы, содержание образования должно быть обогащено принципиально новыми дисциплинами (независимо от профиля подготовки), включая системный анализ и информатику как средство организации и использования проблемно-ориентированных баз знаний; моделирование и оптимизацию; теорию организации человеческих коллективов; социальную психологию и психологию личности. По-

скольку в повседневной практике этого нет — нельзя ожидать, что кризис в образовании будет преодолен сам собой. Косметическими мерами, связанными с добавлением «новинок» в традиционные курсы, здесь не обойтись.

Одновременно следует признать, что появление в образовании новых информационных технологий, использующих компьютерную и видеотехнику, средства коммуникации, предоставляет нам огромный шанс в преодолении кризиса. Оперативная организация проблемно-ориентированных баз знаний на основе реализации структурно-функциональных предметных и межпредметных связей, возможности моделирования процессов, ранее отнесенных к категории сложных, — их главные достоинства. Достоинства, которыми мы, к большому сожалению, пока практически не располагаем.

За всем этим — огромные усилия по созданию нового поколения учебников на основе современных информационных технологий, формирование нового учителя, техническое переоснащение учебного процесса.



М. БУРГИН

канд. физ.-мат. наук

Г. СТЕПЕНКО

канд. пед. наук НИИ педагогики УССР

## Информационный поиск и компьютерная грамотность

В разных ситуациях компьютерная грамотность может иметь разные значения. Так же, например, как в обычном понимании грамотность школьника и грамотность поэта или грамотность журналиста означают существенно разные вещи. Становится все более ясным, что обучение программированию не должно быть главным компонентом компьютерной грамотности в общеобразовательной школе. Дело усугубляется еще и тем, что как считает целый ряд специалистов, в развитии традиционных языков программирования типа Фортрана, Бейсика или Паскаля наступил кризис; следовательно, обучая школьников писать программы на таких языках и тем самым прививая им определенный стиль программистского мышления, школа может оказывать ориентированной только на сегодняшний день, тогда как сегодняшним школьникам предстоит жить и трудиться в обществе завтрашнего дня.

Но что, если не программирование, должно занять центральное место при обеспечении компьютерной грамотности в общеобразовательной школе?

При ответе на этот вопрос следует исходить из того, какие основные задачи стоят перед вычислительной техникой и как эти задачи сочетаются с основной целью общеобразовательной школы — подготовкой молодежи к самостоятельной жизни и труду, воспитанием сознательных строителей нового общества.

Главной задачей внедрения вычислительной техники, наряду с автоматизацией различных производственных процессов, является расширение интеллектуальных возможностей человека. К настоящему времени скорость роста накопленного человечеством знаний достигла величин, заставивших говорить об информационном кризисе. В результате, как считает член-корреспондент АПН СССР В. Г. Болтянский [1], «в век ин-

форматики изменяется само понятие обучения: усвоение знаний уступает место умению пользоваться информацией». Именно информационный взрыв, характерный для современной эпохи, ставит перед системой образования целый ряд сложных проблем. Поэтому в понятие компьютерной грамотности следует включать как основное умение получать нужную информацию с помощью ЭВМ.

К числу средств общения с ЭВМ относятся не только технические средства (клавиатура, монитор, канал связи и др.); не менее важны языки, служащие для взаимодействия человека с машиной. Многие специалисты считают, что главным (а некоторые полагают, что и единственным) средством такого взаимодействия являются языки программирования. На самом деле так было только в первый период использования электронной вычислительной техники. Сейчас же имеется много других языковых средств. Это языки запросов, языки моделирования, электронные таблицы, языки заданий, языковые средства пакетов прикладных программ (в том числе и обучающих программ) и т. п. В последнее время все более реальной становится возможность взаимодействия с компьютером на естественном языке, либо на языке той области, к которой относится решаемая задача, либо на языке деловой прозы. Компьютеры 5-го поколения должны обладать интеллектуальными возможностями, с ними смогут общаться неподготовленные люди. Это означает, что языки программирования хотя и будут нужны, но только для специалистов в области программирования, а массовые пользователи (именно их должна готовить общеобразовательная школа) будут иметь более простые средства взаимодействия с ЭВМ.

Важное место среди таких средств занимают языки запросов, использующие-

ся для извлечения информации из хранилищ, называемых базами данных. Основная структурная единица хранения информации в вычислительных системах — файлы. Этот термин происходит от английского слова file (карточка, досье). В программировании файл означает поименованную последовательность записей. Все записи в файле имеют одинаковое строение и содержат какие-то данные. Пример файла и входящих в него записей приведен в табл. 1. Файлы предназначены для хранения данных ограниченного объема, используемых непосредственно в некоторой программе. Обычно файлы хранятся на внешних запоминающих устройствах ЭВМ (типичным примером которых являются дисководы с магнитными дисками). При пересылке и обработке файлы рассматриваются как единое целое.

Постепенно массивы информации, которые накапливались и хранились многими абонентами (людьми, программами) для долговременного использования, разрослись. Помещать такое количество информации в один файл было не только нецелесообразно из-за неэффективности работы с ним, но часто и вообще невозможно по техническим причинам. Потребовалась организация более сложных систем, в которых файлы могли использоваться в качестве компонентов. Таким образом появились файловые системы, обеспечивающие хранение данных, поиск их на внешних запоминающих устройствах и пересылку в ОЗУ.

Прямой просмотр последовательно расположенных в файлах записей не позволяет добиться высокой эффективности взаимодействия файловых систем с человеком. Улучшить положение можно с помощью введения связей между

различными данными, а также обеспечения дополнительных возможностей взаимодействия (например, поиск и выдача избирательной информации). Реализация этих приемов привела к появлению компьютерных систем, которые стали называться автоматизированными базами данных (БД). Они обычно включают различные аппаратные и программные средства ЭВМ, используемые для хранения и манипулирования данными. Такие БД, как правило, опираются на определенную файловую систему, обеспечивающую выполнение простейших операций с данными. Примером неавтоматизированной БД может служить каталог любой библиотеки. Он является системой, представляющей ряд свойств книг — фамилии авторов, название, год издания, издательство, количество страниц и т. п. БД создаются для хранения информации об определенной предметной области, в нашем примере — о совокупности книг библиотеки.

Современные БД являются автоматизированными и создаются на основе вычислительной техники. Общее управление в них осуществляется специально предназначенной для этого системой управления базой данных (СУБД), состоящей из языковых, алгоритмических и программных средств. Она выполняет функции создания и введения управляемой ею базы, в частности обеспечивает совместное использование данных многими пользователями. Существуют и БД индивидуального пользования. Особенно много их появилось в связи с распространением ПЭВМ. Однако любой такой базой данных могут пользоваться и другие абоненты. При этом обычно предполагается, что данные из одной БД взаимосвязаны, предназначены для одного или нескольких типов приложений

Таблица 1

	Страна	Площадь территории	Население	Столица	Население столицы	Часть света
1.	СССР	22.4 млн. км <sup>2</sup>	276 млн.	Москва	9 млн.	Европа, Азия
2.	США	9.4 млн. км <sup>2</sup>	230 млн.	Вашингтон	750 тыс.	Северная Америка
3.	Англия	244 тыс. км <sup>2</sup>	57 млн.	Лондон	8 млн.	Европа
4.	Индия	3.3 млн. км <sup>2</sup>	650 млн.	Дели	4.8 млн.	Азия
5.	Япония	372 тыс. км <sup>2</sup>	115 млн.	Токио	12 млн.	Азия

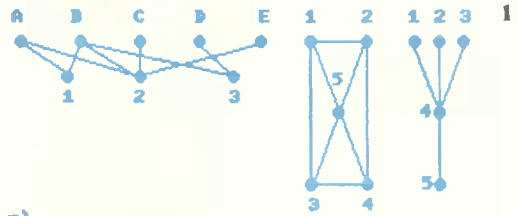
и хранятся так, чтобы быть независимыми от использующих их программ.

Все БД могут быть разделены на три основных типа: реляционные, сетевые и иерархические. Название типа определяется тем, с помощью каких структур представлена информация. В реляционных базах данных основной структурой является отношение. Отношения изображаются в виде двумерных таблиц (пример — табл. 1).

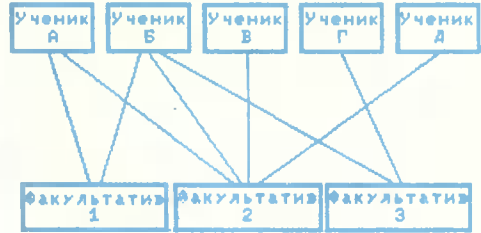
Таблицы, представляющие отношения в реляционных базах данных, обладают следующими свойствами: 1) каждая строка описывает определенный объект (в примере — страну); 2) в таблице нет двух одинаковых строк; 3) каждый столбец содержит определенную однородную информацию, т. е. все элементы столбца имеют одинаковую природу (например, все элементы четвертого столбца табл. 1 — названия городов); 4) каждый столбец имеет имя, описывающее тот вид информации, который содержится в нем (четвертый столбец табл. 1 имеет имя «население»); 5) в операциях, выполняемых с таблицей, строки и столбцы можно просматривать в любом порядке независимо от их содержания.

Такое представление данных имеет целый ряд преимуществ. Оно понятно пользователю, не являющемуся специалистом в области программирования, позволяет легко присоединять новые описания объектов и их характеристики, обладает большой гибкостью при обработке запросов. Большим достоинством реляционных структур является работанность множества операций над ними, с помощью которых строятся алгебры и исчисления отношений, т. е. математические системы, позволяющие точными методами описывать и изучать реляционные БД и происходящие в них процессы.

В сетевых БД основной структурой являются сети, т. е. произвольные графы, в вершинах которых записана та или иная информация, а ребра соответствуют связям между ними. Напомним, что граф — это математическая конструкция, состоящая из вершин и ребер. Каждое ребро соединяет две вершины. Примеры графов приведены на рис. 1, а. В сетевой структуре любая вершина может быть



⇒



⇒

связана с любой другой. Пример сетевой структуры данных приведен на рис. 1, б.

В иерархических БД основной структурой представления информации являются деревья. Обычно дерево определяется как связный граф, не имеющий циклов. Графическое изображение дерева имеет следующий вид. Все вершины разбиты на уровни. На самом высшем (первом) уровне находится только одна вершина, которая называется корнем дерева. Она соединяется ребрами со всеми вершинами, находящимися на втором уровне, и только с ними. Вершины второго уровня соединяются с вершинами третьего уровня ребрами так, что каждая вершина третьего уровня соединена только с одной вершиной второго уровня, и т. д. Пример иерархической структуры в виде дерева приведен на рис. 2.



Если данные имеют естественную иерархическую структуру, то использование иерархических БД имеет много преимуществ. В первую очередь это эффективное использование памяти ЭВМ и простота выполнения многих операций по преобразованию данных.

Вообще, при сравнении между собой трех типов БД можно сделать следующие выводы:

1) с помощью отношений можно выразить сетевые структуры, а значит, и иерархические. Однако делается это за счет повторения одинаковых данных. В результате менее экономно используется память ЭВМ. Например, сетевая структура, изображенная на рис. 1, б, представляется с помощью табл. 2;

Таблица 2

	Факультатив 1	Факультатив 2	Факультатив 3
Ученик А	1	1	0
Ученик Б	1	1	1
Ученик В	0	1	0
Ученик Г	0	0	1
Ученик Д	0	1	0

2) использование отношений во много упрощает работу с БД, делая их доступными для неподготовленных или не очень опытных пользователей;

3) обработка реляционных структур оказывается наиболее простой, хотя и в некоторых случаях менее эффективной, чем структур других типов;

4) существование развитого математического аппарата в теории отношений обеспечивает возможность глубокого теоретического исследования функционирования и особенностей реляционных БД;

5) сетевые и иерархические структуры оказываются удобнее при реализации. Их использование позволяет экономить память и в ряде случаев ускоряет обработку данных.

На основе автоматизированных баз данных создаются автоматизированные информационно - поисковые системы (АИПС), в связи с чем появилась необходимость переводить информационные запросы на язык, понятный ЭВМ. Удобство извлечения информации с помощью АИПС в значительной степени зависит

от того, насколько прост «понятный» ей язык.

Часто понятия базы данных и информационно-поисковой системы используются в одном и том же смысле. Однако между ними существуют определенные различия. Если БД в основном исполняет функции накопления и хранения информации, предоставляя ее для других программ, то ИПС обслуживает непосредственно пользователей.

Банком данных называют более крупную систему того же назначения. Банк данных часто является компонентом автоматизированной системы и обычно состоит из нескольких БД, находящихся под общим управлением.

Вычислительные машины и связанные с ними системы очень быстро развиваются. Одним из свидетельств этого является наблюдаемый в настоящее время переход от обработки данных к оперированию со знаниями. Это, в свою очередь, приводит к появлению банков и баз знаний, которые являются хранилищами знаний.

Согласно современным представлениям, знания от данных отличаются в первую очередь тем, что знания включают интерпретацию данных, т. е. показывают, к чему относятся эти данные и что они означают. Так, если из табл. 1 исключить самую верхнюю строку, то оставшаяся часть будет содержать только данные. Еще одно существенное отличие знаний от данных заключается в их структурированности; иначе говоря, знание всегда образует некоторую систему.

Базы данных, а сейчас и базы знаний входят как неотъемлемая часть в информационно-логические и информационно-вычислительные системы, которые помимо накопления и выдачи хранящейся в них информации выполняют и другие функции. Информационно-логические системы дополнительно обрабатывают выдаваемую информацию, выполняя логические операции, а информационно-вычислительные системы предназначены для решения расчетных задач, разработки программного обеспечения и т. п.

БД и построенные на их основе ИПС могут найти разнообразные применения в процессе обучения: обеспечение ин-

формацией об успеваемости, общественной деятельности, интересах и способностях учеников; составление планов и сценариев уроков; формирование таблиц опорных сигналов; подбор задач и заданий для контрольных и лабораторных работ и т. д.

Сейчас ИПС постепенно становятся интеллектуальными. Для этого в их состав включают специальные аппаратные и программные средства, при помощи которых пользователь общается с ЭВМ на языке, близком к естественному.

В тех случаях, когда пользователь взаимодействует с ЭВМ не прямо, а передает свои задачи оператору, который непосредственно выполняет их на ЭВМ, запрос подается на специальной анкете со стандартными графами, а ответ выдается на стандартных бланках, которые заполняются компьютером по четким правилам. В последние годы пользователи, как правило, непосредственно взаимодействуют с ЭВМ (главной причиной этого послужило появление ПЭВМ), используя либо специальные языки запросов, либо определенное подмножество обычных естественных языков (получившее название деловой прозы). В таких ИПС поисковые процедуры становятся более мощными. С их помощью можно находить в БД ответы на запросы, выраженные не прямо, а косвенно или даже неявно. Например, пользователь БД, изображенной с помощью табл. 1, может задать вопрос: является ли столица США большим городом? Для ответа на этот вопрос машина должна знать, что такое большой город. Это может быть задано следующим условием: город называется большим, если его население больше 5 млн. человек. Имея эту информацию и проведя сравнение населения Вашингтона с указанной в условии величиной, машина выдаст ответ: «Нет». Если хорошо продумать форму подачи запросов и ответов, то это позволит сделать общение между пользователем и системой внешне похожим на диалог на обычном человеческом языке.

ИПС, используемые в настоящее время, делятся на документальные, фактографические и информационно-логические. Документальные информационно-

поисковые системы содержат тексты, в которые иногда включены таблицы и простейшие изображения (графики). Тексты могут быть рефератами статей, статьями, отчетами, патентами и другими документами; отсюда и название — документальные.

Фактографические ИПС содержат данные в виде таблиц, схем, фактов, т. е. в виде структурированной информации. Они обеспечивают поиск и выдачу только той информации, которая в них непосредственно содержится. Если же дополнительно производится логическая обработка информации (как в рассмотренном примере с большим городом), то ИПС относится к типу информационно-логической. В последнее время в связи с широким распространением обработки изображений с помощью компьютеров появились специальные ИПС и БД графической информации.

19

О роли ИПС и БД говорит и то, что в последнее время вычислительные системы начинают превращаться в информационно-вычислительные системы, неотъемлемой частью которых становятся БД, а в число основных функций помимо решения расчетных задач входят накопление информации в БД и выдача ее пользователям.

К тому же и современные тенденции в развитии языков программирования таковы, что структура ряда новых языков (так называемых языков логического программирования) строится по образу и подобию баз данных. Программа в этом случае представляет собой некоторое состояние базы данных, которое служит спецификацией решаемой задачи и содержит определенные программистом отношения. Для вычисления по такой программе компьютеру нужно дать запрос к этой базе данных. В запросе указывается, что же нужно получить. Таким образом, язык программирования содержит и язык запросов, и язык описания состояния некоторого типа баз данных.

Наиболее распространенным из языков программирования нового типа является Пролог, структурно подобный реляционным базам данных. С ним связывают новый стиль в программировании, новый тип программистского мышления.

Предлагаемые в этом языке принципы организации вычислений параллельны по своей сути и потому дают новую и многообещающую точку зрения на возможную архитектуру вычислительных систем [2], что особенно важно в плане разработки компьютеров 5-го поколения.

Основным средством взаимодействия большинства пользователей с ИПС, базами и банками данных и знаний являются языки запросов (ЯЗ).

Так же как и языки программирования (ЯП), ЯЗ предназначены для формулирования задач обработки данных и выдачи нужной информации. Отличаются ЯЗ от ЯП, во-первых, тем, что записи на ЯП — программы — содержат указания для ЭВМ, в которых сказано, что требуется сделать и (до какой-то степени) как это следует сделать; записи на ЯЗ — запросы, — как правило, указывают только, что требуется получить, но не определяют, каким способом. Что и как делать, «знает» специальная программа, с помощью которой обрабатываются запросы и выдаются ответы на них. Такая программа обработки запросов входит обычно в программное обеспечение ИПС систем управления базами данных.

Во-вторых, программа на ЯП в общем случае служит заданием для ЭВМ на получение такой информации, которой в ней нет, а запрос на ЯЗ является заданием на извлечение информации, которая чаще всего содержится в памяти вычислительной машины.

В-третьих, ЯЗ и их применение, как правило, намного проще, нежели ЯП и их применение. Сами запросы, как правило, намного проще, чем даже сравнительно небольшие программы.

Примерами запросов к БД, представленной в табл. 1, могут служить следующие выражения: «Сколько людей живет в СССР?», «Столицей какого государства является Дели?», «Где находится Токио?», «В каких странах живет более 200 млн. человек?»

Ответами на эти вопросы будут соответственно: «276 млн. человек», «Индия», «Япония», «Азия», «СССР, США, Индия».

Приведенные запросы могут быть записаны и по-другому, в зависимости от

того, как устроен ЯЗ. Например, для первого запроса возможны такие формы: «Выдать население СССР» или «Население СССР?» или запись на бланке, изображенном в табл. 3.

Таблица 3

Условие поиска	Что требуется
СССР	Население

Следует отметить, что ЯЗ в первую очередь предназначены для взаимодействия с ЭВМ пользователей-непрофессионалов. Профессиональные же пользователи при работе с БД применяют не обычные ЯЗ, а специальные языки описания данных и языки манипулирования данными.

В силу всего этого именно знание тех или иных ЯЗ и, главное, умение ими пользоваться должно стать основным компонентом компьютерной грамотности для большинства пользователей ПЭВМ, а изучение ЯП и овладение навыками и умениями программиста не должно входить в задачи курса ОИВТ в общеобразовательной школе.

В заключение отметим, что нельзя сводить задачи курса ОИВТ только к овладению компьютерной грамотностью. Поступать так — все равно что сводить задачи современного курса физики к умению использовать автомобиль. Значение информатики для школьников не в этом (хотя это тоже важно), а в том, что она, с одной стороны, дает представление об очень важных сторонах материального мира и человеческого общества, рассматривая их в информационном плане, а с другой стороны, обеспечивает человека мощными средствами (и не только техническими), которыми нужно овладеть для того, чтобы действительно стать всесторонне развитой личностью. Понимание этого привело к тому, что в последнее время стали говорить уже не о компьютерной, а об информационной грамотности [3]. По-видимому, компьютерную грамотность следует считать частью информационной. Это означает, что задача овладения компьютерной грамотностью не снимается с повестки дня, но ее нужно считать только подзадачей более общей за-

20

дачи — овладения информационной грамотностью.

В информационную грамотность входят следующие компоненты.

1. Умение понимать информацию, представленную разного рода текстами (повествовательными, инструктивными, вопросительными) на обычных или научных языках. Это (вместе с требованием второго пункта) означает знание различных языков (как минимум одного естественного языка, элементарного математического языка, языка взаимодействия с ЭВМ и т. д.).

2. Умение выражать информацию о различных объектах с помощью текстов так, чтобы это было понятно определенному адресату (например, другому человеку или ЭВМ). В силу широкого понимания того, что такое текст, это требование включает довольно много. Так, построение математической модели некоторого объекта (процесса) — это выражение информации об этом объекте (процессе) посредством текста на математическом языке, а разработка алгоритма решения какой-либо задачи — это представление информации о решении этой задачи текстом на

алгоритмическом языке. Умение выражать информацию вопросительными текстами, по существу, означает умение формулировать задачи.

3. Умение переводить тексты с одного языка на другой. Примером такого перевода может служить написание программы по имеющемуся алгоритму.

4. Умение для достижения поставленной цели организовывать и планировать действия как отдельных лиц (в частности, свои действия) и машин (в частности, ЭВМ), так и целых систем (человеческих коллективов, многомашинных комплексов, программных средств и т. п.).

Все это и должно стать главным в плане решения прикладных задач курса информатики в общеобразовательной школе.

## Л и т е р а т у р а

1. Завтрашние проблемы? // Юность. 1986. № 10.

2. Борщев В. Г. Пролог — основные идеи и конструкции // Прикладная информатика. 1986. Вып. 2.

3. Велихов Е. П. Новая информационная технология в школе // Информатика и образование. 1986. № 1.

## Как увидеть идею

Нередко рутинная работа, погасив творческий порыв художника, лишает мир нового шедевра. Особенно часто это происходит в архитектуре, где огромный объем чертежных работ, планирования и расчетов до такой степени связывает проектировщиков, что в результате при наличии прекрасной и талантливой идеи получается типовая постройка, в которой лишь автор может угадать черты первоначального замысла.

Освободить зодчих от грустных мыслей о воплощении своих идей в готовые чертежи может пакет программ «Соната» для цветной графической станции «Ирис-4», с помощью которого можно не только быстро «набросать» с почти фотографическим качеством общий вид здания, но и, постепенно детализируя проект, «входить» в его внутренние помещения, осматривать их с разных точек зрения,

перемещать, изменять пропорции и масштаб. Система позволяет оценить и внешний вид не существующего пока здания при различном дневном и искусственном освещении в предполагаемом для его размещения ландшафте.

Все остальное — дело вычислительной техники: конструктивные расчеты и проектирование основы, сетевое планирование

строительства и выдача рабочих чертежей. И все это — точно, быстро и без единой ошибки.

## Снова о пчелах

Широко сообщалось о том, что создано устройство, включающее солнечную батарею, датчик, компьютер и передатчик и в то же время достаточно компактное, чтобы его можно было надеть на пчелу. Устройство передает на землю данные о состоянии организма летящей пчелы.

Исследователи пошли дальше. Недавно создана механическая «пчела» — робот, способный имитировать действия пчелы, сообщаящей своим подругам о местонахождении богатых нектаром участков. Робот управляется компьютером, способен ползать и махать крыльями. Настоящие пчелы понимают его сообщения, хотя и хуже, чем «речь» настоящих пчел.

ЧТО?  
МОЖЕТ  
ЭВМ

# ШКОЛЬНАЯ ЛОКАЛЬНАЯ СЕТЬ 0011

## **Моноканальная однородная школьная локальная вычислительная сеть на базе персональных ЭВМ «Электроника БК-0011»**

является системой технических и программных средств, обеспечивающей обработку и обмен информацией, необходимой для обучения информатике и основам вычислительной техники в школах.

Все ЭВМ в сети подключаются параллельно через контроллеры моноканала к витой паре проводов.

Любое РМУ (рабочее место ученика) может быть преобразовано в РМП (рабочее место преподавателя) установкой на контроллере моноканала абонентского номера 1.

Встроенные в ПЗУ сервисы пользовательского уровня предназначены для обеспечения обмена информацией в школьном классе, и все необходимое вводится для их выполнения в интерактивном **ДИАЛОГОВОМ** режиме.

Встроенные сервисы, используемые учениками со своих рабочих мест как под управлением драйверно-мониторной системы БК-0011, так и под управлением Бейсик-системы, **позволяют:**

*прочитать файл с внешнего запоминающего устройства на дискетах (ВЗУ), подключенного к РМП;*

*прочитать файл с автозапуском с ВЗУ РМП;*

*записать файл на ВЗУ РМП;*

*передать сообщение длиной в 32 символа на служебную часть экрана любого абонента сети.*

Контроллер моноканала в виде отдельного модуля (размером 100×100 мм) подключается к общей шине при помощи разъема на задней стенке микро-ЭВМ БК-0011.

**Встроенные сервисы РМП позволяют преподавателю:**

*просматривать на своем экране содержимое экранов РМУ (время выполнения — 4 с);*

*записывать содержимое области памяти РМП в память РМУ;*

*читать содержимое области памяти РМУ в память РМП;*

*передавать текстовые сообщения на служебную часть экрана любого абонента.*

Под управлением оперативной системы РАФОС в РМП запускается программа обслуживания запросов на передачи файлов, которая помимо встроенных сервисов дает возможность преподавателю передать на РМУ файл с ВЗУ, передать на РМУ файл с ВЗУ с автозапуском.

При разработке использован принцип открытых систем, где выполняется информационная связь между одинаковыми уровнями протоколов обмена в разных абонентских системах посредством сервиса, предоставляемого более низкими уровнями.

В витой паре один провод является сигнальной линией, второй — общий.

Сигнальная линия представляет собой проводное ИЛИ и в концах через согласующие резисторы 160 Ом подключается к источнику питания +5 В.

В сети используются индивидуальный, групповой и широковещательный способы адресации.

Количество абонентов — до 31.

Абонентские системы в сети подразделяются на одно рабочее место преподавателя (РМП) и от 1 до 30 рабочих мест учеников (РМУ).

РМП должно иметь абонентский номер 1.

Номер абонентской системы устанавливается переключками в контроллере моноканала во время установки сети.

**За справками обращаться:  
142500, Московская обл., г. Павловский Посад,  
завод «Экситон», ОНТЭИ. Телефон: 2-26**



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

## Информатика: циклы и вспомогательные алгоритмы

В предыдущей статье мы разобрали методику изучения темы «Ветвления». В этой статье мы разберем методику изучения еще двух тем алгоритмизации: «Циклы» и «Вспомогательные алгоритмы». Напомним, что методика изложения теоретического материала в основном приведена в [2]. Поэтому основное внимание мы концентрируем на методике решения задач. Будет уместно напомнить и о том, что задачи из [1] можно условно разбить на четыре типа (см. [2]): найти ошибку в алгоритме, программе, модели и т. д.;

определить, каков результат выполнения алгоритма (программы);

составить алгоритм (программу);

построить математическую модель, составить алгоритм, написать программу.

I. В теме «Циклы», как и в предыдущей теме «Ветвления», решение многих задач сопровождается составлением протоколов исполнения алгоритмов. При записи протокола для циклического алгоритма естественно фиксировать выполнение или невыполнение условия цикла, указывать, к какому действию переходить, если это условие выполняется, и к какому — если не выполняется. Естественно также отмечать переход к проверке условия цикла после окончания тела цикла (признаком окончания тела цикла служит специальный указатель, например: «Конец цикла»). Необходимо обратить внимание учащихся, что к действию, расположенному за указателем «Конец цикла», исполнитель может перейти только непосредственно после проверки условия продолжения цикла (а, скажем, не после указателя «Конец цикла»).

При решении задач на составление циклических алгоритмов следует обсуждать с учащимися необходимость использования циклов. Целесообразно задавать учащимся вопросы типа: «Почему для составления алгоритма решения задачи недостаточно линейного способа организации действий и ветвлений?», «Какую последовательность действий надо повторять циклически?» (или «Какие действия входят в тело цикла?»), «Каково условие продолжения цикла?», «Где поставить указатель конца цикла?» и т. п.

Для решения одной и той же задачи цикл можно организовать по-разному. Важно, чтобы учащиеся понимали, что выбор последовательности действий, повторяющихся в цикле, зависит от выбора условия продолжения цикла и наоборот. Часто бывает необходимо до начала цикла (или после его окончания) исполнить некоторые из действий, входящих в тело цикла, иначе задача будет решена не полностью (см., например, ниже задачу 2). В ряде задач несколько действий перед циклом нужны для его подготовки: без них оказывается невозможной первая проверка условия цикла. Так, в вычислительных алгоритмах, как правило, перед исполнением цикла нужно присвоить некоторой переменной начальное значение. Одна из типичных ошибок состоит как раз в том, что учащиеся забывают это сделать.

При изучении темы «Циклы» впервые начинает существенно проявляться различие между использованием переменных в математике и информатике. В курсе математики учащиеся привыкли, что в процессе решения задачи каждое новое значение переменной обозначается по-своему. Скажем, хотя в уравнении неизвестная обозначается буквой  $x$ , для каждого корня вводится свое обозначение:  $x_1, x_2, \dots$ . В алгоритме же, предна-

наченном для нахождения этих корней Вычислителем, каждый из них может быть обозначен буквой  $x$  и выдан на экран по мере исполнения алгоритма. В предыдущих темах учитель мог при желании обойти это различие. Ведь в алгоритмах, не содержащих циклов, каждое действие выполняется не более одного раза, поэтому для каждого присваивания можно ввести свою переменную. Изучая циклические алгоритмы, это различие обойти невозможно: если тело цикла содержит присваивание, то при исполнении алгоритма одной и той же буквой будут последовательно обозначаться разные значения.

Это обстоятельство создает дополнительную трудность при решении задач. В ча-

диалог «Учитель—Ученик» при составлении циклических алгоритмов для Вычислителя.

1. Что рисует Чертежник, выполнив следующий алгоритм из исходного положения, показанного на рисунке?

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



*Решение.* Приведем протокол выполнения алгоритма.

Пока впереди край, повторять:  
 Повернуть налево.

Условие выполняется. Исполняем действие:



Конец цикла.  
 Пока впереди край, повторять

Переходим к проверке условия продолжения цикла. Проверяем условие «Впереди край».

Условие не выполняется.

Цикл закончен.

Выполняем действия, записанные после слов «Конец цикла».

Сделать шаг.



Пока впереди край, повторять:

Условие не выполняется.

Цикл закончен.

Выполняем действие, записанное после слов «Конец цикла».

Сделать шаг.



стности, поэтому задачи на составление циклических алгоритмов сложнее задач, которые учащиеся решали ранее. Наш опыт показывает, что создание алгоритма становится для школьников существенно более простым делом, если вначале они излагают «нечеткий» план этого алгоритма. Такой план обязательно должен включать в себя распределение «ролей» между переменными, значения которых будут меняться в цикле. Можно сказать, что «нечеткий» план алгоритма подобен программке балетного спектакля, где указаны роли, названы исполнители и рассказан сюжет.

Мы разберем три типичные задачи на тему «Циклы». Первая задача иллюстрирует построение протокола выполнения Чертежником циклического алгоритма; вторая — идею подготовки к циклу; третья — нечеткую запись, распределение «ролей» переменных и

2. Составьте алгоритм нахождения фальшивой монеты среди настоящих монет того же достоинства с помощью чашечных весов, если известно, что фальшивая монета тяжелее настоящей.

*Разбор решения.* Если на каждой чашке весов лежит по одной монете, то либо весы находятся в равновесии, либо одна из монет перетянет. Ясно, что в первом случае обе монеты настоящие, а во втором случае более тяжелая монета — фальшивая.

Поэтому, чтобы найти фальшивую монету, надо положить одну (произвольную) монету на какую-нибудь, скажем левую, чашку весов и по очереди сравнивать с ней остальные монеты.

Значит, сначала надо выполнить действия:

Положить монету на левую чашку весов.

Положить монету на правую чашку весов.

Затем необходимо циклически повторять действия:

Убрать монету с правой чашки весов.

Положить следующую монету на правую чашку весов.

Эти действия составляют тело цикла. Выполнив цикл надо, пока весы уравновешены.

Текст алгоритма:

Положить монету на левую чашку весов.

Положить монету на правую чашку весов.

Пока весы уравновешены, повторять:

Убрать монету с правой чашки весов.

Положить следующую монету на правую чашку весов.

Конец цикла.

Если монета, лежащая на левой чашке весов, тяжелее, чем монета, лежащая на правой чашке весов, то:

Предъявить монету с левой чашки весов:

Иначе:

Предъявить монету с правой чашки весов.

Конец ветвления.

Типичная ошибка школьников: они забывают положить монету на правую чашку весов до начала цикла. В результате весы оказываются неуровновешенными, и тело цикла выполняться не будет. В качестве ответа будет выдаваться первая попавшая монета, которую мы положили на левую чашку весов.

3. Составьте для Вычислителя алгоритм, пользуясь которым он запросит последовательность из 10 чисел и определит, является ли она арифметической прогрессией.

*Разбор решения.* Решение задачи существенно зависит от того, какой выбран критерий для определения, является ли данная последовательность арифметической прогрессией. Как правило, учащиеся вспоминают одно из двух: либо определение арифметической прогрессии, либо формулу общего члена. Мы разберем здесь только решение, опирающееся на формулу общего члена, поскольку оно проще (в [3] разбирается и второе решение).

Во время составления алгоритма решения задачи может состояться следующий диалог между учителем и учащимися.

**Вопрос.** Как распознать арифметическую прогрессию с помощью формулы общего члена?

**Ответ.** Последовательность  $a_n$  образует арифметическую прогрессию, если для каждого натурального  $n$  выполняется равенство:

$$a_n = a_1 + (n-1) \cdot d, \quad (1)$$

где  $d$  — разность прогрессии.

**Вопрос.** Как определить  $d$ ?

**Ответ.** Чтобы определить  $d$ , достаточно знать первые два члена прогрессии  $a_1$  и  $a_2$ . Тогда  $d = a_2 - a_1$ .

**Вопрос.** Допустим, что Вычислителю известны первые два члена данной последовательности (а значит, и их разность).

Как он должен действовать, чтобы выяснить, образуют ли прогрессию члены этой последовательности?

**Ответ.** Вычислитель должен запрашивать один за другим члены последовательности, начиная с третьего, и каждый раз проверять, удовлетворяет ли запрошенное число формуле общего члена арифметической прогрессии (т. е. формуле (1)). Если да, то анализ последовательности надо продолжить; если же нет, то Вычислитель должен сообщить, что данная последовательность не является прогрессией, и прекратить выполнение алгоритма (например, по команде СТОП).

Этот диалог в достаточной степени готовит учащихся к «нечеткой» записи требуемого алгоритма. Когда такая запись сделана, целесообразно обсудить использование переменных в алгоритме.

**Вопрос.** Какие переменные Вычислителю понадобятся и каковы их «роли»?

**Ответ.** Нечеткая запись алгоритма подсказывает, что в нем понадобятся следующие «действующие лица»:

«первый член последовательности» (обозначим его буквой  $x$ );

«второй член последовательности» (обозначим его буквой  $y$ );

«претендент на роль разности прогрессии» (обозначим его буквой  $d$ );

«очередной член последовательности» (обозначим его буквой  $z$ );

«номер очередного члена последовательности» (обозначим его буквой  $n$ ).

**Вопрос.** Какие действия нужно повторять циклически?

**Ответ.** В цикле нужно запрашивать очередной член последовательности ( $z$ ); проверять, удовлетворяет ли он формуле (1); если нет, то останавливаться, если да — увеличивать номер  $n$  на единицу.

**Вопрос.** Каково условие продолжения цикла?

**Ответ.** Цикл надо заканчивать, когда все десять членов последовательности рассмотрены. Значит, условие продолжения цикла:  $n \leq 10$ .

**Вопрос.** Какая подготовка требуется для цикла?

**Ответ.** Сначала Вычислитель должен запросить первый и второй члены последовательности ( $x$  и  $y$ ), найти их разность ( $d$ ) и присвоить  $n$  начальное значение, равное 3.

**Вопрос.** В каком случае Вычислитель должен сообщить, что данная последовательность является арифметической прогрессией?

**Ответ.** Если для всех членов последовательности выполнено условие (1).

Вопрос. Значение какой переменной показывает, сколько членов последовательности уже проверено?

Ответ. п.

Вывод: При  $n=11$  (если, конечно, это значение будет достигнуто) все члены последовательности проверены и удовлетворяют условию (1), поэтому тело цикла больше исполнять не нужно и Вычислитель должен сообщить: «Последовательность является арифметической прогрессией».

После этого учащиеся без особых затруднений записывают алгоритм решения задачи:

Запросить  $x, y$ .

Присвоить  $d$  значение  $y-x$ .

Присвоить  $n$  значение 3.

Пока  $n \leq 10$ , повторять:

Запросить  $z$ .

Если  $z \neq x + (n-1) \cdot d$ , то:

Сообщить: «Последовательность не является арифметической прогрессией».

26

Стоп.

Конец ветвления.

Присвоить  $n$  значение  $n+1$ .

Конец цикла.

Сообщить: «Последовательность является арифметической прогрессией».

После разбора этого решения можно предложить учащимся подумать, как уменьшить количество используемых переменных (можно «заставить»  $z$  играть роль  $y$ ).

II. Отладка алгоритмов, содержащих вспомогательные алгоритмы, имеет свои особенности. В первую очередь надо отладить вспомогательные алгоритмы (составив несколько протоколов их исполнения). После этого, составляя протокол выполнения основного алгоритма, выполнение вспомогательного алгоритма надо рассматривать как одно допустимое действие, фиксируя только конечный результат.

Все задачи на составление алгоритмов с помощью вспомогательных алгоритмов можно подразделить на три группы:

а) выделение вспомогательных алгоритмов из готовых алгоритмов (как в «четкой», так и в «нечеткой» записи);

б) составление алгоритмов с помощью готовых вспомогательных алгоритмов;

в) составление алгоритмов методом пошаговой детализации.

Проводя аналогию с конструированием, можно сказать, что группа а) соответствует пониманию того, что любая сложная конструкция состоит из отдельных деталей; группа б) соответствует сборке конструкции из готовых деталей; группа в) соответствует полному циклу конструирования — от создания необходимых деталей до окончательной сборки.

Отметим, что первая из этих групп,

хотя и не применяется при решении задач, несет важную методическую нагрузку — она является подготовкой к решению задач из остальных двух групп, служит лучшему усвоению самого понятия «вспомогательный алгоритм».

При решении задач группы б) целесообразно задавать учащимся вопросы типа: «Почему алгоритм решения задачи проще составить с использованием вспомогательных алгоритмов?», «Какую последовательность действий надо включить в состав вспомогательного алгоритма?», «Каковы аргументы и результаты вспомогательного алгоритма?», «Как оформить вспомогательный алгоритм?», «Как записать команду вызова вспомогательного алгоритма?» и т. п.

При решении задач группы в) вопросы могут быть следующими: «На какие подзадачи разбивается данная задача?», «Каковы требования к вспомогательным алгоритмам решения этих подзадач?», «Что следует включить в заголовки этих вспомогательных алгоритмов?», «Каков основной алгоритм решения задачи?», «Являются ли алгоритмы решения подзадач достаточно простыми? Следует ли продолжить дробление некоторых из них на более мелкие подзадачи?»

Приведем разбор решений трех задач — по одной из каждой группы.

1. В отрасли внедрен хозрасчет — от бригады до министерства. Бригада из своей прибыли делает отчисления предприятию, предприятие — главу, а глава — министерству. Чтобы подсчитать, какую сумму отчислений с прибыли бригады получит министерство, был составлен следующий алгоритм:

Запросить  $s, b, p, l$ .

Присвоить  $t$  значение  $b+s \cdot p$ .

Присвоить  $m$  значение  $s-l$ .

Если  $m < 0$ , то:

Сообщить «Отчисления равны нулю».

Стоп.

Конец ветвления.

Присвоить  $z$  значение  $m \cdot t$ .

Присвоить  $t$  значение  $b+z \cdot p$ .

Присвоить  $m$  значение  $z-l$ .

Если  $m < 0$ , то:

Сообщить «Отчисления равны нулю».

Стоп.

Конец ветвления.

Присвоить  $z$  значение  $m \cdot t$ .

Присвоить  $t$  значение  $b$ .

Присвоить  $m$  значение  $z-l$ .

Если  $m < 0$ , то:

Сообщить «Отчисления равны нулю».

Стоп.

Конец ветвления.

Присвоить  $z$  значение  $m \cdot t$ .

Сообщить  $z$ .

В этом алгоритме — три почти одинаковых

куска. Составьте соответствующий вспомогательный алгоритм, а затем запишите основной алгоритм.

*Разбор решения.*

Нетрудно заметить, что если отбросить два действия — первое и последнее, то наш алгоритм разбивается на три похожих блока по 7 строк в каждом: сначала два присваивания, затем ветвление и еще одно присваивание. Чем отличаются эти блоки? Первый и второй — лишь наименованием переменных: там, где в первом куске  $s$ , во втором —  $z$ . А третий блок отличается еще и отсутствием второго слагаемого в первой команде присваивания. Однако можно считать, что это слагаемое присутствует, просто  $p$  в третьем блоке равно нулю. Таким образом, можно записать вспомогательный алгоритм следующим образом:

- Присвоить  $t$  значение  $b+s \cdot p$ .
- Присвоить  $m$  значение  $s-l$ .
- Если  $m < 0$ , то:
- Сообщить «Отчисления не производятся».
- Стоп.
- Конец ветвления.
- Присвоить  $z$  значение  $m \cdot t$ .

Но для того чтобы этот вспомогательный алгоритм можно было вызвать из другого алгоритма, надо записать его заголовок, т. е. указать название, аргументы и результаты. Какие из переменных, используемых в этом алгоритме, берутся им извне? Ответ ясен: это переменные  $b, s, p, l$ . Остальные переменные ( $m$  и  $z$ ) вычисляются в самом алгоритме. Вот как может выглядеть заголовок вспомогательного алгоритма:

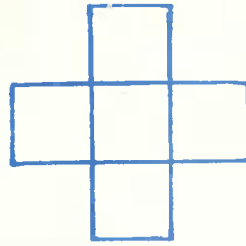
- Алгоритм «Отчисления».
- Аргументы:  $s, b, p, l$ .
- Результат:  $z$ .

Теперь нетрудно записать основной алгоритм:

- Запросить  $s, b, p, l$ .
- Выполнить алгоритм «Отчисления», взяв в качестве аргументов  $s, b, p, l$ .
- Выполнить алгоритм «Отчисления», взяв в качестве аргументов  $z, b, p, l$ .
- Выполнить алгоритм «Отчисления», взяв в качестве аргументов  $z, b, 0, l$ .
- Сообщить  $z$ .

Обратите внимание на то, что при втором и третьем вызове алгоритма «Отчисления» в качестве первого аргумента указывается  $z$ : результат первого вызова вспомогательного алгоритма является аргументом для второго вызова, а результат второго вызова — аргументом для третьего. (Это соответствует «движению» прибыли: прибыль бригады является основой для определения отчислений главку, а доход главку является основой для определения отчислений министерству.)

2. Составьте для Чертежника алгоритмы рисования фигуры, изображенной на рисунке:



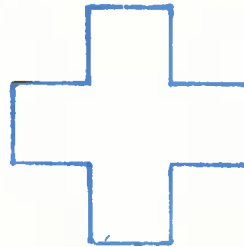
*Разбор решения.*

Для рисования фигуры будем использовать вспомогательный алгоритм, разобранный в тексте учебника [1]:

- Алгоритм «Скобка».
- Повернуть налево.
- Повернуть налево.
- Сделать шаг.
- Повернуть налево.
- Сделать шаг.
- Повернуть налево.
- Сделать шаг.

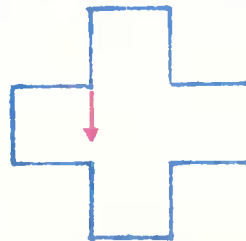
Там же на основе алгоритма «Скобка» построен алгоритм рисования следующей фигуры:

27



Вот этот алгоритм:

- Выполнить алгоритм «Скобка».
  - Повернуть налево.
  - Выполнить алгоритм «Скобка».
  - Повернуть налево.
  - Выполнить алгоритм «Скобка».
  - Повернуть налево.
  - Выполнить алгоритм «Скобка».
- Оформим этот алгоритм как вспомогательный. Для этого достаточно как-то назвать его, например: «Крест». Воспользовавшись этими вспомогательными алгоритмами, легко написать основной алгоритм. Выполнив алгоритм «Крест», Чертежник будет находиться в таком положении:



(стрелка показывает положение Чертежника после исполнения записанной последова-

тельности действий). Теперь ему остается лишь нарисовать квадрат внутри фигуры. Текст основного алгоритма:

Выполнить алгоритм «Крест».  
Повернуть налево.  
Повернуть налево.  
Выполнить алгоритм «Скобка».  
Повернуть налево.  
Сделать шаг.

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

Мы рассмотрим лишь в общих чертах схему возможного обсуждения этой задачи, весьма богатую, как нам представляется, в учебном отношении. Прежде всего, отметим, что она допускает разные пути решения и обсуждать эти пути как раз удобнее всего, составляя макро-блок-схемы, отражающие пошаговую детализацию задачи.

Первое решение, которое обычно предлагается учащимся, основано на следующей идее: среди трех заданных чисел найти максимальное и минимальное и выдать в качестве ответа оставшееся число. Таким образом, задача разбивается на три подзадачи.

Выбрать максимальное из трех запрошенных чисел.

↓  
Выбрать минимальное из трех запрошенных чисел.

↓  
Из трех запрошенных чисел выбрать число, не совпадающее ни с максимальным, ни с минимальным числом.

Выбор максимального из трех чисел в свою очередь может быть далее детализован (фактически это проделано в объяснительном тексте учебника [1]). По аналогии разбирается и второй блок — выбор минимального числа. Дальнейшая детализация третьего блока очевидна:

Первое запрошенное число сравнить с максимальным и минимальным.

↓  
Второе запрошенное число сравнить с максимальным и минимальным.

↓  
Третье запрошенное число сравнить с максимальным и минимальным.

Фактически каждый из этих блоков предназначен для ответа на один и тот же вопрос: не совпадает ли некоторое число с одним из двух заданных чисел (в данном случае — с максимумом или минимумом)? При составлении соответствующего вспомогательного алгоритма потребуются использование сигнальной переменной (см. п. 3г из [4]).

Детализацию на этом можно считать законченной. Если буквально следовать этой детализации, то понадобится три вспомогательных алгоритма: «Выбор максимального из трех чисел», «Выбор минимального из трех чисел» и «Сравнение числа с двумя заданными числами». И в целом алгоритм, реализованный на данной идее решения, получается довольно громоздким. (Можно, правда, для поиска минимального числа воспользоваться вспомогательным алгоритмом поиска максимального: для этого надо у аргументов поменять знак на противоположный, затем найти максимальное число и у результата снова поменять знак, но и это ненамного улучшит положение.)

Вторая идея решения задачи может состоять в том, чтобы сначала упорядочить первые два из трех запрошенных чисел, а затем определить место третьего числа по отношению к этим двум. Этой идее соответствует следующая детализация:

Упорядочить первые два из запрошенных чисел.

↓  
Определить место третьего числа: если третье число меньше минимального, то сообщить это минимальное число; если третье число больше максимального, то сообщить это максимальное число; если оба предыдущих условия не выполняются, то сообщить третье число.

Вспомогательный алгоритм «Упорядочение двух чисел» может выглядеть так:

Алгоритм «Упорядочение двух чисел».

Аргументы:  $x, y$ .

Результаты:  $m, n$ .

Если  $x < y$ , то:

Присвоить  $m$  значение  $x$ .

Присвоить  $n$  значение  $y$ .

Иначе:

Присвоить  $m$  значение  $y$ .

Присвоить  $n$  значение  $x$ .

Конец ветвления.

Второй блок без особого труда реализуется с помощью ветвлений. Можно, однако, воспользовавшись вспомогательным алгоритмом «Упорядочение двух чисел», сделать запись основного алгоритма более изящной:

Запросить  $a, b, c$ .

Выполнить алгоритм «Упорядочение двух чисел» при  $x=a, y=b$ .

Если  $c > m$ , то:

Выполнить алгоритм «Упорядочение двух чисел» при  $x=c, y=n$ .

Конец ветвления.

Сообщить  $m$ .

Третье решение задачи может быть построено на следующей идее: сначала расположить запрошенные числа по возрастанию, а затем предъявить второе по счету число. Наша задача разбивается на две подзадачи:

Расположить числа по возрастанию.

Предъявить второе по счету число.

Вторая из этих подзадач решается одним действием Вычислителя (действием «Сообщить»). Решение первой подзадачи оформим как вспомогательный алгоритм. Выскажем требования к этому вспомогательному алгоритму. Исходными данными для него являются три числа:  $x$ ,  $y$ ,  $z$ . Результатом будет последовательность  $a$ ,  $b$ ,  $c$  — эти же цифры в порядке возрастания.

Основной алгоритм будет выглядеть так:

Запросить  $x$ ,  $y$ ,  $z$ .

Выполнить алгоритм «Упорядочение трех чисел», взяв в качестве аргументов  $x$ ,  $y$ ,  $z$ .

Сообщить  $b$ .

Существует довольно много различных способов упорядочения (сортировки) чисел. Для составления алгоритма «Упорядочение трех чисел» мы воспользуемся методом «пузырька» (методику изучения метода «сортировки» см., например, [5]). Представим себе, что числа находятся в трех ящиках с номерами 1, 2, 3. Упорядочить числа в соседних ящиках — это означает либо поменять их местами (если это необходимо), либо оставить на прежних местах. По методу «пузырька» для упорядочения всех трех чисел нужно упорядочить числа в первом и втором ящиках, затем — во втором и третьем и наконец — в первом и

втором. Таким образом, задача упорядочения трех чисел разбивается на три подзадачи:

Упорядочить числа в первом и втором ящиках.

Упорядочить числа во втором и третьем ящиках.

Упорядочить числа в первом и втором ящиках.

Для решения всех этих подзадач достаточно одного вспомогательного алгоритма упорядочения двух чисел (см. выше). Роль ящиков, как обычно, сыграют три переменные.

## Литература

1. Основы информатики и вычислительной техники / А. Г. Гейн, В. Г. Житомирский, Е. В. Линецкий и др. Свердловск: УрГУ, 1989.

2. Гейн А. Г., Житомирский В. Г., Линецкий Е. В., Сапир М. В., Шолохович В. Ф. Программно-методический комплекс по курсу школьной информатики // Информатика и образование. 1988. № 3, 4.

3. Дидактические материалы по курсу информатики. Гл. 4—6 / А. Г. Гейн, М. В. Сапир, В. Ф. Шолохович и др. Свердловск: СвГПИ, 1989.

4. Гейн А. Г., Линецкий Е. В., Сапир М. В., Шолохович В. Ф. Информатика: алгоритмические конструкции // Информатика и образование. 1989. № 5.

5. Фукс Л. Алгоритмы поиска и сортировки // Информатика и образование. 1988. № 2.

М. ГОЛЬЦМАН, А. ДУВАНОВ, Я. ЗАЙДЕЛЬМАН, Ю. ПЕРВИН

## Информация вокруг нас

В Роботландии выделяются четыре основных педагогических направления: *информационное, компьютерное, алгоритмическое и творческое*. Естественно, что все они органически переплетаются в содержании обучения и в методике.

В приводимых описаниях уроков сохранены только фрагменты, относящиеся к информационному направлению (их заголовки в планах выделены).

Урок 2.

Тема: Информация в нашей жизни.

План урока

Обсуждение газетных вырезок и рассказов из домашних заданий.

Проверка знаний техники безопасности, правил включения и выключения машин.

Информация вокруг нас.

Примеры учителя, примеры учеников.

Как понять, что такое информация.

Почему трудно определить информацию.

Одно- и многократное нажатия клавиш управления курсором.

Правила игры в Мудрого Кролика.

Практическая работа: игра-программа Мудрый Кролик.

Домашнее задание: Придумать примеры информации.

### § 1.1. Что такое информация

На перемене перед уроком информатики Петя Кук подошел к одноклассникам:

— Ребята! У меня есть информация!

— Где же она у тебя?

— Вот, — и Петя достал из портфеля листок бумаги, на котором почерком директора школы была написана записка.

— Какая же это информация? Это записка Андрея Петровича!

— А вы прочитайте.  
Хором ребята прочитали:

*«Ребята! Завтра в наш город приезжают гости — школьники из Ленинграда. Они будут у нас в школе и придут в ваш класс на урок информатики.»*

*Андрей Петрович».*

— Да, новость интересная! Но почему ты, Петя, назвал эту новость информацией?

— А как же! Это ведь сообщение для нас.

— Тогда и у меня есть информация. Вот, пожалуйста: завтра днем температура воздуха будет от 10 до 12 градусов тепла.

— А вчера хоккеисты «Крыльев Советов» победили «Спартак». Это информация или не информация?

— Конечно, информация. Всякие сведения — это информация.



— Да? А что же такое сведения?

Этот коварный вопрос Катя задала Пете Куку, начавшему разговор. Кук задумался.

— Сведения... гм... ну, это информация.

— Сам запутался,— засмеялась Катя.— Давайте лучше спросим учителя.

И как только прозвенел звонок, Катя подняла руку:

— Сан Саныч! Мы хотели узнать: что такое информация? Но у нас получилось, что информация — это сведения, а сведения — это информация. Заколдованный круг какой-то!

— Неудивительно. Вы задали сложный вопрос. Дело в том, что нельзя определить информацию, то есть закончить предложение, начатое словами «Информацией называется...». Чтобы это понять, попробуйте объяснить, например, что здесь нарисовано (рис. 1).

— Это дом!

— Из каких фигур составлен этот рисунок?

— Из двух квадратов и из треугольника.

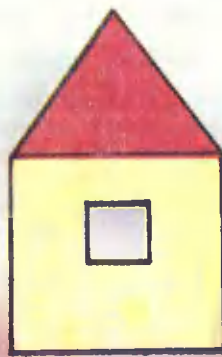
— Верно! А из чего состоят треугольник и квадрат?

— Из линий.

— А из чего состоят линии?

— Из точек.

— А что такое точка? Трудный вопрос,



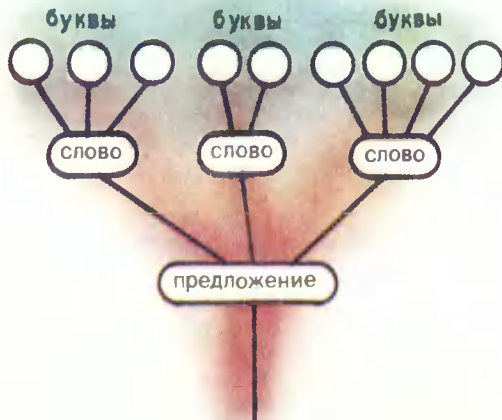
правда? Такой же трудный, как вопрос об информации. А вот еще вопрос. Объясните: что такое предложение?

— Это несколько слов, объединенных общим смыслом.

— Не совсем точное, но для нашей задачи подходящее определение. А что такое слово?

— Это несколько букв, написанных в правильном порядке.

— А что такое буква? Это тоже вопрос трудный. Вы заметили, что при объяснении сложной вещи или сложного правила всегда используют более простые вещи или правила (рис. 2). Так, объяснение фигуры сведено к линии, а линии — к точке. Объяснение предложения сведено к словам, а слов — к буквам. В конце цепочки объяснений оказываются такие понятия, которые объяснить через более простые уже нельзя. Например, точку невозможно свести к чему-нибудь более простому. Основным понятием





является и информация. Оно не сводится к более простым. Достаточно уметь приводить примеры информации и понимать, что с нею можно делать. А ваше пояснение, что информацию можно представлять как сведения, совершенно правильное. Итак, ваши примеры информации?

— Записка Андрея Петровича! Она содержит сведения о предстоящем событии.

— Таблица умножения: сведения о правилах умножения чисел.

— План нашего города. Если на него посмотреть, то получишь сведения о том, как добраться до нужного места.

Информацию не надо пытаться точно определять, но очень важно получить от детей множество примеров. С этой целью учитель генерирует один-два примера, а затем приглашает учеников к поиску собственных примеров. Работая со сложным понятием, полезно научить младшего школьника оперировать примерами исследуемого понятия и уметь применять его раньше и вместо того, чтобы заучивать строгое определение.

Важно показать учащимся, что информация может восприниматься любым из органов чувств человека. Более того, примеры из мира фауны — обоняние у собаки, осязание у крота, слух у кошки — подтверждают детям объективность существования информации в материальном мире.

## § 1.2. Как человек получает информацию

Человек способен воспринимать информацию по-разному. У человека пять разных органов чувств (рис. 3):

**зрение;** глазами люди различают цвета, воспринимают зрительную информацию, к которой относятся и текстовая, и числовая, и графическая;

**слух;** ушами воспринимается звуковая информация — речь, музыка, звуковые сигналы, шум;

**обоняние;** носом люди получают информацию о запахах окружающего мира;

**вкус;** с помощью языка можно получить информацию о том, каков предмет на вкус — горький, кислый, сладкий, соленый;

**осязание;** кончиками пальцев, на ощупь, можно получить информацию о температуре предмета — горячий он или холодный, о качестве его поверхности — гладкий или шершавый.

Урок 3.

Тема: Работа с информацией.

План урока

Обсуждение домашнего задания — примеров информации.

Что можно делать с информацией. Примеры учителя.

Обсуждение примеров учащихся.

Выделение видов информационных процессов: иницирующий пример учителя, примеры учеников.

Обсуждение рецепта приготовления манной каши.

Рецепт манной каши — план решения задачи.

Практическая работа: игра-программа Мудрый Кролик.

Домашнее задание: план приготовления манной каши.

В начале урока надо проверить умения школьников генерировать примеры информации. Вероятно, примеров будет много. Учитель выделяет наиболее удачные, фиксирует активность учащихся и качество их ответов.

Главная задача — рассмотреть операции, которые обычно осуществляются над информацией: хранение, передача и обработка. В учебнике этой теме посвящен § 1.3. Это описание историй, которые случились с Петей Куком. Первая из них в учебнике подробно проанализирована. Поэтому и саму историю, и ее анализ можно предложить ученикам на уроке. Обсуждая историю, учитель отмечает новые слова-термины: хранение, обработка, передача.

## § 1.3. Что можно делать с информацией?

*История 1.* Петя взял в школьной библиотеке книгу о кошках. Он прочитал, что были случаи, когда кошку увозили за сотни километров и она все равно возвращалась назад. Петя понял, что кошки очень хорошо запоминают дорогу. В своем читательском дневнике Петя записал эту мысль.

Обсуждение первой истории:

1) Какую информацию получил Петя Кук? Кук узнал о случаях, когда кошки возвращались, будучи увезенными далеко от дома. Эта информация представлена в текстовом виде.

2) Где находилась эта информация? Местом хранения информации была библиотечная книга.

3) Как Петя получил эту информацию? Для передачи информации (из книги в память) Петя Кук воспользовался глазами. У этой передачи информации есть точное

3





32

направление: из книги — в Петину голову.

4) Где оказалась информация, когда Петя ее получил? Новое место хранения информации — Петина память.

5) Какое заключение сделал Кук, прочитав книгу? Результат обработки полученной информации — новая информация о способностях кошек запоминать дорогу. Эта обработка информации сделана в Петинной голове.

6) Что сделал Петя с новой мыслью? Он записал ее в свой дневник. Произошла передача информации (рис. 4).

**История 2.** На уроке математики Кука вызвали к доске. Учитель попросил его решить пример:  $(2+3) \cdot 4$ . Петя, конечно, справился с этой задачей. А потом, вернувшись на свою парту, задал себе вопросы:

1) Как попала ко мне информация, нужная для решения задачи?

2) Где хранилась информация, пока я решал задачу?

3) Что происходило с информацией во время решения?

4) Где хранился результат после решения?

5) Как результат стал известен учителю и всему классу? Как была выведена информация?

Кук нашел ответы на все вопросы. А вы?

В обеих историях можно выделить хранение информации, передачу информации, обработку информации. Хранение, передача и

обработка — это информационные процессы. Все, что происходит с информацией, всегда включает хранение, передачу и обработку.

Рассмотрев подробно пример с информацией о кошках, учитель просит учащихся придумывать собственные примеры. Обилие примеров позволит выделить наиболее удачные. Подводит черту потоку примеров домашнее задание — § 1.3. Выполняя задания параграфа, надо придумать новые примеры работы с информацией.

Обсуждение домашнего задания не окончено. Оно завершается задачей о рецепте приготовления манной каши. Но эта задача связана уже с другим направлением курса — алгоритмизацией. Этой и другим близким задачам, на которых школьники знакомятся с понятием алгоритма, посвящен урок 4.

Урок 5.

Тема: Работа с информацией, хранение.

План урока

Обсуждение домашнего задания — формального описания алгоритма перевозки волка, козы и капусты через реку.

Игра в классе.

Лабораторная работа: Перевозчик.

Примеры хранения информации. Обсуждение примеров детей.

Информационные носители: от древнего мира до наших дней.

Способы хранения. Примеры.

Практическая работа: Диалог с машиной по программе Привет.

Домашнее задание: Придумать примеры хранения информации.

Вопрос на повторение вводит тему сегодняшнего занятия: «Что можно делать с информацией?» Из высказываний учеников следует выделить главное: три основных компонента информационного процесса — хранение, передача и обработка. Сегодня подробнее рассматривается первый из них — хранение.

Рассказ можно вести по тексту § 1.4. Примеры на хранение информации, аналогичные примерам § 1.4, дети смогут воспроизвести сами в достаточном количестве. Новое понятие — информационный носитель. Предложив ученикам несколько примеров, надо попросить их продолжить перечень.

### § 1.4. Хранение информации

Если спилить дерево, то по кольцам на стволе можно определить, сколько ему лет, дождливым или засушливым был каждый год его жизни и многое другое. Значит, в дереве хранится информация обо всей его жизни.

Давным-давно, когда на Земле жили первобытные люди, возникла необходимость хранить различные сведения о способах охоты, рыбной ловли, земледелия. Для этого люди использовали рисунки, зарубки на палках, узелки на веревках. По рисункам древних

4





людей мы узнаем, как они жили.

Заучивая стихи, человек хранит их в своей памяти. Иногда, не полагаясь на память, он записывает информацию в записную книжку или тетрадь. Например, расписание уроков и домашнее задание школьник хранит в дневнике.

Современный человек придумал новые способы хранения информации: на кинолентах, магнитных лентах, грампластинках.

Устройства, на которых хранится информация, называются **информационными носителями**. Всем знаком информационный носитель — книга. Дверной косяк, на котором родители ежегодно отмечают рост своего ребенка, — тоже информационный носитель. Информация, нужная для работы компьютера, записана на магнитном диске. Диск — это тоже информационный носитель.

Урок 6.

Тема: Хранение информации.

План урока

Обсуждение домашнего задания.

Организация хранения информации: записная книжка, спортивный протокол, индекс-указатель. Исправление ошибок, замена символов.

Лабораторная работа: Тренажер Правилка.

Домашнее задание: Предложить способ хранения книг в домашней библиотеке.

В § 1.5 учебника продемонстрированы различные способы хранения (упорядочения) информации. Главный вывод этого параграфа — хранение данных следует организовать так, чтобы максимально удобным (эффективным) оказался их поиск.

§ 1.5 — предмет подробного изложения на уроке при активном участии класса. К каждому из предложенных способов упорядочения информации у учителя должен быть подготовлен хотя бы один дополнительный пример. Полезно, если наряду с примерами из практики человеческого опыта хранения информации найдутся примеры организованного хранения информации в природе (хранение наследственных признаков растениями

и животными, хранение истории Земли в геологических отложениях).

Чтобы подчеркнуть обыденность задач организации информации, школьникам надо предложить дискуссию о способах, которыми можно упорядочить книги в домашней библиотеке.

### § 1.5. Хранить, чтобы искать

Вы уже знаете, что иметь дело с хранением информации человеку приходится очень часто. Но информацию недостаточно просто сохранить, надо сделать так, чтобы потом, когда она понадобится, ее можно было быстро найти. Для этого люди придумали различные способы организации хранения информации. Вот как Петя Кук познакомился с некоторыми из этих способов.

*История 1.* Петя Кук был летом в пионерском лагере. У него появилось много новых друзей: Таня Петухова, Вова Абрамов, Дима Силкин. Друзья дали ему свои адреса. Как Пете лучше сохранить эту информацию?



Решение простое: Пете надо завести записную книжку. В записной книжке для каждой буквы отведена своя страничка. Такие странички, помеченные буквами-указателями, следуют в том же порядке, что и буквы в алфавите: сначала — страничка А, потом — страничка Б и так далее, до последней странички с буквой Я. Первым в свою записную книжку Петя записывает Вову Абрамова, потому что фамилия Вовы начинается с первой буквы алфавита. В середине книжки, на странице с буквой П размещается адрес Тани Петуховой. А немного дальше, на странице С, — адрес Димы Силкина. Если Петя хранит информацию именно так, ему легко найти нужные сведения: ведь Кук хорошо знает алфавит.

Этот способ хранения информации используется очень часто. По алфавиту расположены слова во всех словарях и энциклопедиях, фамилии и названия учреждений в телефонном справочнике, города в расписании движения самолетов.

*История 2.* Как найти нужное место в большой книге? Можно, конечно, просто перелистывать страницы, пока не откроется нужная, но этот способ занимает много вре-

мени. Гораздо быстрее посмотреть в содержание. В содержании обычно перечислены названия всех глав книги и номера страниц, на которых расположены эти главы. Отыскать нужную главу в содержании намного проще, чем искать ее по всей книге.

Не всегда удается найти нужную информацию, пользуясь только содержанием. Например, в этом учебнике о хранении информации говорится не только в § 1.4, который так и называется — «Хранение информации», но и в § 1.3, § 1.5 и даже в § 4.11, который вы еще не читали. Для облегчения поиска информации в книгах, особенно научных, часто используют индекс. В индексе перечислены по алфавиту (опять алфавитное хранение!) все основные понятия, которые есть в книге, и указано, в каких разделах или даже на каких страницах можно узнать о них. Индекс есть и в нашем учебнике. Чтобы найти в учебнике материал о хранении информации, надо найти в индексе слово «хранение» и посмотреть, какие параграфы там указаны. Затем нужно посмотреть в содержании номера страниц, на которых напечатаны эти параграфы, найти эти страницы и прочитать их.

*История 3.* У Пети Кука неприятность: он натер на ноге мозоль новым ботинком и не смог бежать в школьном осеннем кроссе, в котором участвовал весь класс. Учитель физкультуры попросил Петю записать информацию о всех бегунах. Петя уселся около столика на финише. «Вот задача! Как их записать?» — подумал он. Учитель посоветовал Пете: «Пиши фамилии ребят в столбик в том порядке, как они прибегут на финиш: тогда всех перепишешь и заодно будет видно, кто кого обогнал».

#### *Вопросы и задания.*

1. В этом параграфе описаны разные способы хранения информации. Какой из них использован для записи учеников в классном журнале? А какой лучше применить для составления списка родителей, пришедших на родительское собрание?

2. Как хранятся почтовые марки в вашей коллекции?

3. Около домашнего телефона Петя поставил коробку с карточками. На карточках записаны телефоны Петиних знакомых. Как вы посоветуете Пете расставить эти карточки?

4. В расписании самолетов указано, в какой город и в какое время вылетает каждый рейс. В каком порядке надо расположить эту информацию, чтобы было удобно пассажиру, покупающему билет? А как будет удобнее диспетчеру аэропорта?

5. В каком параграфе учебника вы узнаете, что такое исполнитель? На какой странице находится этот параграф?

#### *Урок 7.*

**Тема:** Передача информации.

#### *План урока*

**Обсуждение домашнего задания:** порядок в домашней библиотеке.

**Обсуждение примеров передачи информации.**

**Игры в передачу информации.**

Постановка задачи о Ханойской башне.

Решение задачи о Ханойской башне с двумя кольцами на детской пирамиде.

Способ формальной записи решения.

Знаки препинания на клавиатуре.

Лабораторная работа: знаки препинания и специальные символы.

Домашнее задание: составить и формально описать алгоритм переноса трех колец Ханойской башки.

Проверка домашнего задания — это впечатления детей от их собственной работы по наведению порядка в домашней библиотеке. Трудно предусмотреть активность детей по этому вопросу. Поэтому в плане урока достаточно отметить опрос двух учеников (не более!), чтобы первая часть урока не показалась классу скучной. Надо попросить школьников, чтобы они ответили не только на вопрос, как они навели порядок в книгах, но и объяснили, почему они выбрали тот или иной метод. Если библиофилы окажутся неактивными, можно спросить о том же юных филателистов: вопрос любителям марок был поставлен в конце § 1.5, заданного на дом.

От предыдущей темы — «Хранение информации» — к теме сегодняшнего урока — «Передача информации» — переход вполне естественный. Достаточно, рассказывая о передаче информации (§ 1.6), трактовать ее как перенос информации с одного носителя на другой. Обратите внимание, что большая часть сюжетов, посвященных передаче информации в § 1.6, иллюстрирует не деятельность человеческого общества, а жизнь мира животных и растений. Этот факт — подтверждение объективности информации, существующей вне и независимо от человеческого сознания. В доступной форме этот тезис следует донести до школьников.

#### **§ 1.6. Передача информации**

Хорошо, когда на складе много товаров. И все же лучше, если товары служат людям, а не лежат без движения на складе. Об информации можно сказать то же самое: чтобы информация была полезной, ее надо своевременно передать.

Информацию передают друг другу животные и растения.

Своими движениями пчелы-разведчицы «рассказывают» соседкам по улью, в какую сторону надо лететь за медом.

В жаркой Индии по берегам рек растут густые заросли растения с удивительным названием «стыдливая мимоза». Когда начинается тропический ливень, стыдливая мимоза спешит свернуть свои листочки, спасая их от сильных струй. Но самое интересное,

что, как только первые капли дождя упадут хотя бы на одно из растений, сигнал о наступающем дожде передается от ветви к ветви, и все растения длинной цепи зарослей сворачивают свои листья.

Деятельность людей связана с передачей информации. Древний способ передачи информации — письмо, отправляемое с гонцом.

В африканских джунглях в старину важные новости передавались от одного племени к другому барабанным боем.

Моряки иногда пользуются для передачи информации флажковой азбукой.

Разговаривая, мы передаем друг другу информацию. Люди придумали разные устройства для быстрой передачи информации: телевизор, радио, телеграф, телефон. К числу устройств, передающих информацию с большой скоростью, относятся ЭВМ. Компьютер может быстро получить информацию, прочитав ее, например, с магнитного диска. Еще быстрее ЭВМ выводит информацию на экран монитора — это тоже передача информации.

В передаче информации всегда участвуют две стороны: передает информацию — источник, а получает — приемник (рис. 5). Например, объясняя урок, учитель передает информацию ученикам. В этом случае учитель — источник информации, а ученики — приемники. А когда ученик отвечает на вопрос, он становится источником, а учитель — приемником.



Хотя материал § 1.6 излагается в доступной форме, все же приводимые примеры достаточно абстрактны. Тема допускает различные нетрадиционные методические приемы. Например, простейшую внутриклассную «почту» из записочек можно на несколько минут легализовать, чтобы продемонстрировать процесс передачи информации. Учитель сам может поучаствовать в этой игре, назвав несколько различных пар учащихся: источник — приемник.

Столь же увлекательной покажется детям передача информации, если научить их хотя бы нескольким буквам азбуки глухонемых (или, удобнее, ее упрощенной «школьной» модификации) и попросить передать несколько простых сообщений-слов из одного угла класса в другой.

После этого обращение к детям с просьбой привести примеры передачи информации, несомненно, поднимет много рук. Надо, чтобы, рассказывая о факте передачи информации, дети называли источник, приемник и носители отправляемой и получаемой информации. Учитель приходит на помощь, поясняя сложные ситуации.

Одна из тем девятого урока — «Искажение информации». Учитель просит школьников назвать примеры передачи информации. Подхватив один из примеров, следует показать, что при этой передаче информация может быть искажена. Далее, обоб-

щая, можно рассказать о важности предохранения информации от искажений при передаче. В этом фрагменте урока могут быть использованы примеры, вопросы и задания из соответствующего параграфа учебника (здесь опущенного).

В конце параграфа поставлен вопрос по поводу игры «Испорченный телефон». Этот вопрос полезно обсудить и даже реализовать. Хорошо, если идею игры выдвинут дети. Тогда учителю останется только поддержать это начинание и организовать игру. Слишком длинная цепочка (весь класс) неудобна. С другой стороны, нельзя делить класс на играющих и неиграющих, так как зрелищная сторона «Испорченного телефона» не впечатляющая. Поэтому ребят надо поднять из-за столов и построить в две цепочки (даже в три, если детей много). Правила игры можно напомнить, хотя обычно детям они знакомы.

Иницирует игру учитель. Единое иницирующее слово для двух (и тем более трех) цепочек придаст игре дополнительный интерес. Важно поддержать высокий темп игры. Комментируя ход игры и поведение игроков, полезно использовать терминологию информационных процессов: «информация», «передача», «источник», «приемник», «помехи».

## Урок 12.

### Тема: Обработка информации.

#### План урока

Проверка домашнего задания: Игра «Живые шахматы».

Проверка домашнего задания: описание алгоритма перестановки коней.

Лабораторная работа: выполнение на компьютере перестановки шести коней.

Примеры обработки информации, их обсуждение и анализ.

Исправление ошибок, смешанные режимы.

Лабораторная работа: режимы «смесь» и «суперсмесь» на тренажере Правилка.

Домашнее задание: составить и записать алгоритм перестановки четырех коней.

В теоретической части урока — продолжение разговора об обработке информации. Примеры прошлого урока рассматривали обработку как изолированный этап информационного процесса. На сегодняшний урок вынесен § 1.8. Его главная цель — показать комплексный характер информационного процесса, в котором обработке отведена роль важного этапа.

## § 1.8 Обработка информации

Чтобы получить новые сведения, необходимо по определенным правилам обработать имеющуюся информацию.

Вот, например, как можно предсказать погоду на завтра. Если вечером громко квакают лягушки, выпала роса, похолодало, на закате небо было чистым, нет ветра и в воздухе чувствуются запахи, можно смело собираться в поход — на следующий день погода будет хорошей. Если комары очень больно кусаются, порывистый ветер, на закате тучи,

низко летают ласточки, то погода будет плохой.

Собирая информацию, которую дает природа, и применяя перечисленные здесь правила, получают информацию о том, какая завтра будет погода.

Информацию, которую обрабатывают, называют исходной. После обработки исходной информации получается новая информация (рис. 6).



Информация обрабатывается людьми, животными и техническими устройствами.

Ученик получает условия задачи (исходную информацию), думает (обрабатывает) и сообщает ответ (новую информацию).

Повар пробует суп. Вкус супа — исходная информация. Затем он думает (обрабатывает) и получает новую информацию — чего в супе не хватает.

36 Служебная собака по запаху ищет человека. Здесь запах — исходная информация, а направление, куда пошел человек, — новая.

Калькулятор производит сложные вычисления. Введенный в него пример — исходная информация, а полученный ответ — новая.

Компьютер — специальный прибор, созданный человеком для обработки информации. Он позволяет обрабатывать информацию очень быстро и точно и именно поэтому является помощником человека и в труде, и в учебе, и в развлечениях.

*Вопросы.*

1. Петя Кук и Катя Пушкиова придумали игру в информацию. Кук, который изображал компьютер, придумывает алгоритм обработки информации, но не рассказывает о нем. Катя сообщает Пете данные; Петя, обработав их, говорит результат. Катя должна отгадать, как Петя обрабатывает информацию.

К а т я. Три медведя.

П е т я. Тре мидвидя.

К а т я. Пришли они в лес.

П е т я. Прешле оне в лис.

Тут Катя воскликнула: «Я догадалась!»

А вы догадались?

2. Пошел котенок Васька в лес позавтракать. Увидел ягоду бруснику. Попробовал скушать — кислая. Учужал гриб боровик, попробовал его — горький. Вдруг слышит шуршание в траве и видит: лежит в густой траве комочек взъерошенный. «Ну, — думает котенок, — тобой и позавтракаю». Хотел он лапой подкатить комочек чуть-чуть поближе, да как закричит: «Ой! Ай!» Комочек ежом оказался. Сел Васька на пенёк и задумался. Думал он, думал и решил, что лучше с ежом не связываться, а поесть кислых ягод да горьких грибов.

Где в этом рассказе говорится о хранении, обработке и передаче информации?

Котенок Васька получает информацию, которая хранилась в ягоде, в грибе, в еже. Для получения этой информации ему пришлось пустить в ход все свои органы чувств: зрение (увидел ягоду), обоняние (учужал гриб), вкус (ощутил кислый привкус ягоды, горечь гриба), слух (услышал шуршание ежа в траве), осязание (потрогал колючие иголки). Правда, об обработке информации в этой истории сказано коротко (но выразительно): «Сел Васька и задумался». Зато очевиден результат обработки — развернутый логический вывод: невкусная ягода и гриб предпочитают колючему ежу.

История про котенка Ваську, несмотря на всю ее простоту, может дать богатый материал для работы в классе. С этой целью можно использовать традиционно-сказочный циклический характер сюжета: в первый раз объектом исследования Васьки служит ягода (увидел — попробовал), во второй раз — гриб (учужал — попробовал), в третий раз — еж (услышал — попробовал). Анализ первой встречи — с ягодой — учитель берет на себя, а описание двух других, пользуясь сказочным подобием ситуаций, поручает детям. Сам учитель ограничивается при этом лишь корректировкой вопросов и ответов. Сказочная цикличность помогает и в формулировке результатов этого сложного информационного процесса с повторяющейся несколько раз обработкой информации: первая обработка дает результат — кислая ягода, вторая — горький гриб, третья — столь неприятные ощущения, что делается вывод о предпочтении невкусных растительных продуктов.

Урок 13.

Т е м а: Поход за информацией.

*План урока*

Прогулка по запланированному маршруту — «Поиск информации» — с коллективным обсуждением ситуаций.

Форма сегодняшнего занятия нетрадиционна. Урок проводится в соседнем лесу. В содержании и методике этого занятия в наибольшей степени отразится специфика школы, которая может быть расположена вдали от леса или даже городского парка. Однако, даже не выходя из школьного здания, можно организовать интересный поход за информацией, посетив в таком походе различные кабинеты, библиотеку, столовую и даже раздевалку. Инициатива педагога плюс фантазия детей позволяют создать интересный сценарий урока.

Здесь урок описан так, как он мог бы проходить в школе, расположенной по соседству с лесом в маленьком районном городке среднерусской полосы.

В поход идет весь класс. Но как и на обычных занятиях в кабинете, ребята разбиваются на две группы. Поэтому у учителя информатики в походе есть помощник — пионервожатый, шеф-старшеклассник или кто-нибудь из родителей.

Подготовка к этому уроку требует от учителя

значительно больше усилий, чем обычно. Ему надо предварительно пройти по маршрутам обеих групп, тщательно их проверить, заготовить и разложить по тайникам записки-информацию, сделать необходимые метки, запастись компасами и лопатками, познакомиться с маршрутом своего помощника и проинструктировать его, предупредить директора школы.

Учитель начинает урок. Распахивается дверь. Появляется человек с запиской интригующего содержания: «Срочно кончайте урок! Мудрый Кролик в беде! Он забыл волшебное слово и не может построить ни одного лабиринта. Волшебное слово найти непросто. Пусть две команды возьмут листки с информацией и срочно отправляются в путь по двум разным маршрутам. Не забудьте взять компасы!»

Вслед за текстами некоторых из приводимых ниже листков информация, которую будут разыскивать школьники) указаны комментарии в скобках. Это пояснения для учителя.

**Листок 1 а** (а — индекс первой группы): Идите к старому дереву. С северной стороны копайте.

**Листок 1 б** (б — индекс второй группы): Идите к выходу на поляну. Направо у двух ближних деревьев копайте.

Дети отправляются по маршруту и находят в указанных местах буквы «п» — первая группа, «м» — вторая группа, а также новые листочки с информацией.

**Листок 2 а:** 25 шагов на юг и 30 шагов на восток (по компасу). Ищите наверху. (Надо поднять головы и поискать на дереве.)

**Листок 2 б:** Расшифруйте сообщение, где поменялись местами две пары букв: Пе трепанко к езору. Пед блажйишам ерохевым кустем ацато. (Здесь поменялись местами «о» и «е», «а» и «и»).

Новый этап поиска дает «о» — первая группа, «я» — вторая группа и следующую порцию информации для дальнейшего поиска.

**Листок 3 а:** Идите вдоль забора. Там, где тропинка уходит от забора, лежат три камня. Ищите. (Записка лежит под одним из камней.)

**Листок 3 б:** 15 шагов на юго-запад (по компасу). Остановитесь у высокой сосны. Найдите на ней дуло. Осмотрите его.

Результаты поиска: «е» — первая группа, «р» — вторая группа.

**Листок 4 а:** Следуйте по тропинке дальше. У опушки леса — муравейник. В  $n$  метрах от него на юг — ямка.

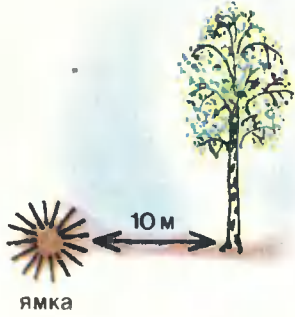
$$n = (24 - 6 \cdot 2) - (14 : (31 - 24))$$

**Листок 4 б:** На листке нарисована картинка (рис. 7).

На этом этапе было обнаружено: «т» — первая группа, «и» и (картинка) еще «и» — вторая группа; ребята из второй группы удивлены — сразу две одинаковые буквы «и»; что бы это значило?

**Листок 5 а:** В двадцати метрах от муравейника на север — мост через ручей. Ищите на дне. (Записка нацарапана на доске, которая положена под камень на дне ручья прямо под мостом.)

**Листок 5 б:** Около домика лесника стоят три



березы. Ваша — самая высокая. (Записка приколата у основания ствола.)

Вот что нашли ребята на сей раз: «к» — первая группа, «н» — вторая группа.

**Листок 6 а:** Идите вдоль ручья к берегу озера. Найдите лодку. (В этом месте единственная лодка. Она перевернута. Под перевернутой лодкой — записка.)

37

**Листок 6 б:** В 50 шагах от колодца вдоль дороги растёт большая береза. На второй снизу сухой ветви жук-древоточец оставил след в виде нужной вам буквы. Ищите внизу. (Листок с дальнейшим маршрутом — в траве точно под указанной веткой.)

Информация этого этапа поиска: «ю» — первая группа, «о» — вторая группа.

**Листок 7 а:** На листке рисунок (рис. 8).

**Листок 7 б:** Дорога пересекает просеку. Слева от дороги на просеке — куст можжевельника. Копайте в двух шагах от него в сторону дороги.

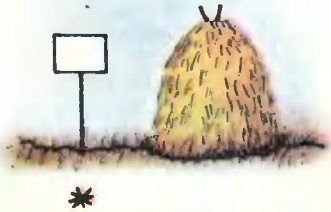
Найденные буквы: «м» — первая группа, «а» — вторая группа.

**Листок 8 а:** Идите по берегу озера по дороге, ведущей к школе. Справа у дороги увидите стог сена. С южной стороны под стогом есть для вас информация.

**Листок 8 б:** Поднимитесь на горку. Название деревни, которую вы увидите с ее вершины, начинается с нужной вам буквы. Следующий листок с информацией о маршруте вам даст вожатый.

В результате этого этапа получено «р» — первая группа, «ф» — вторая группа (деревня называется Федорово, заранее этого сказать было нельзя, так как рядом, но вне поля зрения находится Орехово).

**Листок 9 а:** Под лестницей у входа в школу с левой стороны.



Листок 9 б: Под лестницей у входа в школу с правой стороны.

Ребята уже догадались, конечно, какое волшебное слово им следовало разыскать, и, найдя последнюю букву в указанном месте, составляют слова: первая группа — «компьютер», вторая группа — «информация». Если поход за информацией закончился не только удачно, но и быстро, то остаток урока дети могут провести за игрой в Мудрого Кролика, которому они так помогли.

Урок 15.

Тема: Кодирование информации.

План урока

Задачи о переливании с помощью неградуированных сосудов.

Лабораторная работа: Переливашка.

Обсуждение примеров обработки информации. Выделение этапов кодирования и декодирования. Задачи на шифрование текстов.

Домашние задания: написать дома закодированный рассказ и принятый способ кодировки.

Главный тезис, который надо донести до школьников, — чтобы передать информацию, ее необходимо кодировать. Обсуждение задач кодировки — удобный повод для разговора о взаимосвязи информационных процессов. Хранение, передача и обработка информации не существуют изолированно.

### § 1.10. Кодирование информации

Человек работает у компьютера, набирает на клавишах буквы и цифры. Эта информа-

ция попадает в машину и записывается на магнитный диск. На диске эта информация хранится в виде намагниченных и не намагниченных участков.

Радист получил сообщение, переданное азбукой Морзе. Из точек и тире сложились буквы, из букв — текст.

На соседнем корабле вывесили гирлянду разноцветных флагов. Капитан прочитал это сообщение и передал его телеграммой в штаб флота.

Во всех этих примерах исходная и конечная информация оказалась представленной в разной форме, на разных носителях. Искazилась ли информация? Совсе нет. Изменение информации здесь связано не с искажением, а с кодированием. Кодирование — это преобразование информации в удобную для передачи или хранения форму.

Например, тексты кодируются с помощью букв и знаков препинания. При этом одна и та же информация может быть закодирована по-разному, по-русски, по-английски, по-китайски...

Информация всегда хранится и передается в виде кодов. Нельзя хранить информацию без носителя. Точно так же нельзя хранить и передавать просто информацию: она всегда имеет какую-то форму, т. е. закодирована.

Первая глава учебника завершается двумя параграфами об устройстве ЭВМ. Их главная цель — показать сущность компьютера как 1) универсальной и 2) информационной машины.

*От авторов.* Журнал уже публиковал материалы о программно-методической системе обучения информатике «Роботландия». В этом году журнал будет рассказывать о ведущих темах этого курса.

Методическое обеспечение Роботландии составляют учебник для школьника и книга для учителя. Эти материалы не привязаны ни к точному возрасту учеников, ни к длительности обучения. В зависимости от возраста и формы занятий (урок, факультатив, кружок) на их основе можно построить курсы разной продолжительности и уровня сложности. В журнале будут напечатаны отдельные фрагменты учебника (опущены некоторые параграфы и большинство вопросов и упражнений) и методический комментарий. Надеемся, что эти публикации смогут послужить опорой для проведения занятий по информатике с младшими школьниками.

Учебник состоит из 9 глав, соответствующих основным темам курса:

Глава 1. Информация вокруг нас.

Глава 2. Алгоритмические этюды.

Глава 3. Первое знакомство с компьютером.

Глава 4. Исполнители.

Глава 5. Черные ящики.

Глава 6. Текстовый редактор Микрон.

Глава 7. Графический редактор Раскрашка.

Глава 8. Музыкальный редактор Шарманщик.

Глава 9. Исполнитель Кукарача.

Работа над учебником продолжается, предполагается дополнить его главами, посвященными базам данных, коммуникациям.

Книга для учителя содержит планы и описания возможных вариантов уроков, рассчитанных на 45 мин.



## Методика составления алгоритмов

Основная задача курса программирования в вузе — научить каждого студента решать на ЭВМ достаточно сложные задачи своей области техники (или науки). Аналогично одной из основных задач курса информатики в школе является прививание школьникам навыков программирования. Как же эти задачи решаются в жизни? Практика показывает, что в основном неудовлетворительно. Такой вывод автор делает на основе опыта преподавания упомянутых курсов в школе, в институте усовершенствования учителей, в вузе.

Основное препятствие здесь — сложность и неэффективность обучения процессу алгоритмизации и отладки программ. Возьмем процесс разработки схемы алгоритма задачи, т. е. структуры программ. Это один из наиболее сложных и трудоемких этапов процесса решения задачи на ЭВМ. Однако в учебной литературе по программированию он освещен весьма слабо.

В настоящей работе излагается подход к решению этой задачи, описывается методика составления схем алгоритмов, точнее говоря, предлагается некоторая унифицированная процедура перехода от описания процесса решения задачи в словесно-формульном виде к схеме алгоритма этой задачи, причем такой схеме, которая может быть механически перекодирована в программу на языке программирования.

Особенность методики и ее суть в том, что она позволяет строго математически выводить формулы и логические выражения (условия), составляющие «начинку» схемы алгоритма и программы, поэтому при отладке программы этап обнаружения алгоритмических (семантических) ошибок практически выпадает и отладка резко упрощается.

При обычном же подходе указанные формулы и условия получаются интуитивно, и именно этап выявления семантических ошибок является основным средством подтверждения правильности их методом проб и ошибок.

Указанная методика излагается без излишней формализации и математизации, использование ее в курсах программирования технических вузов и в школьном курсе информатики не вызывает затруднений. Об этом свидетельствует успешный опыт преподавания в школе и в вузе.

Описываемая методика не претендует на теоретическую новизну и является, по существу, эвристической. В основе ее лежит

методика составления алгоритмов с одним циклом.

### Обозначения и определения

Будем различать два типа схем.

1. Подробные схемы описывают процесс решения задачи в терминах операций, допустимых в используемом в дальнейшем языке программирования, т. е. описывают процесс решения со степенью подробности, необходимой в указанном языке.

2. Укрупненные схемы содержат блоки, реализующие операции произвольного вида (такие блоки будем называть укрупненными). Каждый такой блок описывается самостоятельной подробной схемой.

В блоках «процесс» подробных схем будем использовать только выражения вида

$$X := A, \quad (1)$$

где  $X(A)$  — любая переменная (арифметическое выражение).

Выражение (1) выполняется так же, как оператор присваивания в языке программирования. Кроме того, будем использовать рекуррентные выражения вида

$$Y := f(Y). \quad (2)$$

Здесь переменная, стоящая слева (справа) от знака «:=», означает последующее (предшествующее) значение одной и той же величины  $Y$ .

В блоках проверки условий будем записывать только логические выражения, в частности отношения, например:  $X < Y + 2$ .

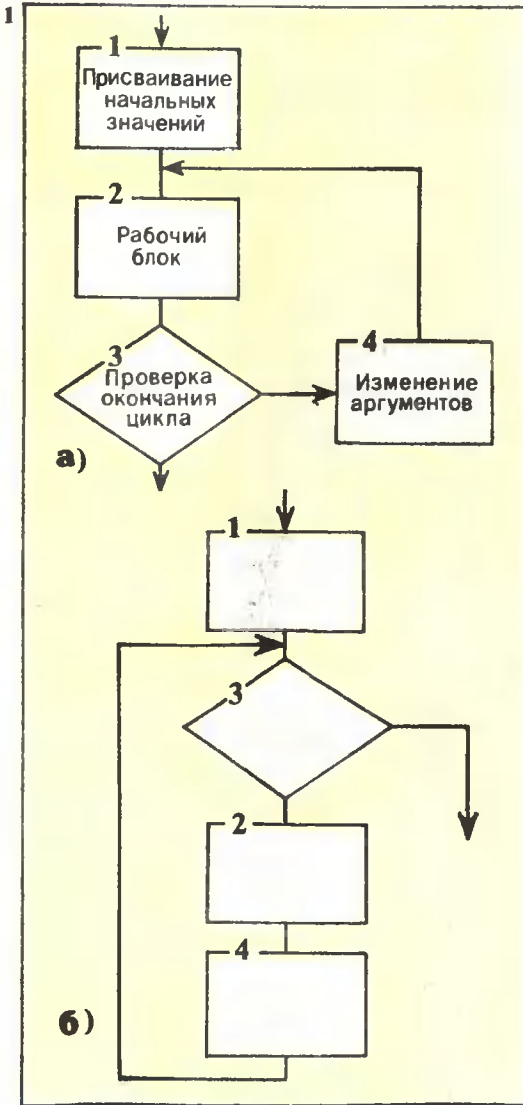
Массивом будем называть множество величин, обозначенных одним именем с различными целочисленными индексами, изменяющимися по порядку (в школьном курсе информатики это понятие обозначено термином «таблица»).

### Типовая схема циклического алгоритма

Структура циклического алгоритма в любом случае может быть описана одной из схем, которые будем называть типовыми, приведенными на рис. 1.

В качестве типовой схемы циклического алгоритма можно выбрать и иную схему, но в любом варианте присутствуют все четыре блока, поэтому выбор типовой схемы не отражается на излагаемых ниже методиках.

Мы отдаем предпочтение схеме вида 1а только потому, что при ее использовании схемы с несколькими вложенными циклами выглядят несравненно проще, чем при использовании схемы 1б.



### Составление алгоритмов с одним циклом

Как известно, циклическим называется алгоритм, в котором получение результата обеспечивается многократным вычислением по одним и тем же формулам при разных значениях входящих в них переменных.

Отсюда следует, что для составления циклического алгоритма необходимо располагать такими формулами, многократное вычисление по которым дает искомый результат. Подобные формулы можно назвать рабочими.

Например, при вычислении

$$Y = \sum_{i=1}^n b_i$$

рабочая формула имеет вид  $Y := Y + b_i$  (при начальном значении  $Y = 0$ ).

Говоря более строго, для однозначного

задания циклического алгоритма необходимо указать три объекта:

множество начальных значений переменных;

множество рабочих формул;

условие окончания повторения цикла (условие ОПЦ) или условие повторения цикла (условие ПЦ).

Если все они заданы в явном виде, то для получения схемы алгоритма задачи достаточно разместить их в типовой схеме циклического алгоритма.

Если же они не заданы в явном виде, то предлагаемая методика требует поэтапного описания процесса решения задачи и вывода на его основе рабочих формул для вычисления результатов.

Из этого же описания и полученных формул выводятся начальные значения всех переменных.

Далее методом математической индукции можно доказать справедливость полученных формул.

Условие ОПЦ (ПЦ) выводится затем из сравнения выражений для некоторой величины на  $i$ -м и последнем этапах.

Для построения схемы все полученные объекты (формулы, начальные значения, условие ОПЦ (ПЦ)) размещаются в типовой схеме алгоритма.

Вывод формул и доказательство справедливости их проиллюстрируем следующим примером.

Определить сумму элементов с нечетными номерами последовательности

$$a_1, a_2, a_3, \dots, a_n,$$

где  $n$  четное.

Описание процесса решения задачи по этапам:

1)  $S_1 = a_1 + a_3;$

2)  $S_2 = S_1 + a_5;$

3)  $S_3 = S_2 + a_7.$

Из сопоставления формул трех этапов не сложно подметить, что будет справедливо

$$i) S_i = S_{i-1} + a_{2i+1} \quad (3)$$

Последний этап:

$$S_r = S_{r-1} + a_{n-1}$$

Приводим операцию 1-го этапа к виду формулы  $i$ -го этапа:

$$S_1 = S_0 + a_3$$

Отсюда следует, что  $S_0 = a_1$ , т. е. найдено начальное значение переменной  $S$ . Кроме  $S$  в формуле (3) используется  $i$  — номер этапа. Начальное значение  $i$ , очевидно, равно 1, закон изменения  $i := i + 1$ . Других переменных нет.

Условие ПЦ вытекает из того, что индекс величины  $a$  в формуле (3) не может превышать  $n-1$ , т. е.  $2i+1 \leq n-1$  и  $i \leq (n-2)/2$ .

Доказательство справедливости выражения (3).

При  $i=1$  выражение верно, что очевидно.

Пусть (3) верно при некотором  $i=k$ ; покажем, что оно верно и для следующего числа  $k+1$ :

$$S_k = S_{k-1} + a_{2k-1} = a_1 + a_3 + \dots + a_{2k-1}$$

$$S_{k+1} = a_1 + a_3 + \dots + a_{2k-1} + a_{2k+1} = S_k + a_{2k+1}$$

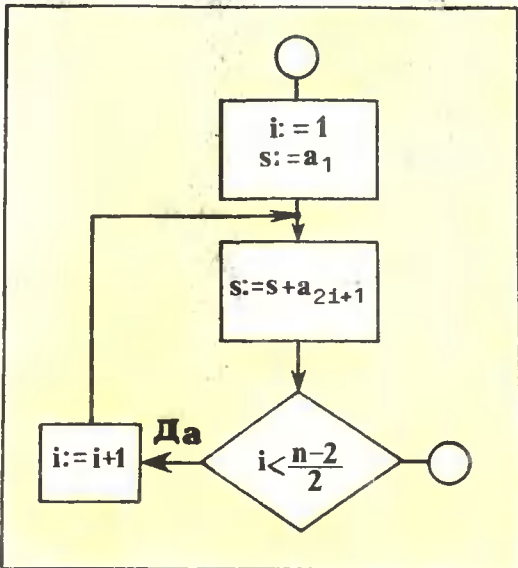
т. е.

$$S_{k+1} = S_k + a_{2(k+1)+1}$$

Мы получили (3), где вместо  $i$  стоит  $k+1$ ; следовательно, на основании аксиомы математической индукции заключаем, что (3) верно для любого натурального числа  $i$ .

Таким образом, мы вывели рабочую формулу для вычисления результата, начальные значения всех переменных и условие ПЦ.

Заполнив типовую схему алгоритма, мы получаем схему алгоритма решения данной задачи (рис. 2), правильность которой доказана.



Очевидно, что данные рис. 2 позволяют написать сразу и программу решения задачи на любом алгоритмическом языке, правильность которой также доказана.

### Методика составления алгоритмов с одним циклом

1. Формулируется условие задачи.

2а. Выделяются исходные данные и выбирается их структура, т. е. определяется, в виде какой совокупности переменных и массивов нужно представить данные задачи.

В простых задачах используется правило: если дано множество значений некоторой величины, закон изменения которой отсут-

ствует, то множество рассматриваем как массив и включаем в исходные данные.

В противном случае массив можно не создавать, а использовать (в целях экономии памяти) рекуррентное выражение.

2б. Выделяем результаты: определяем их количество; назначаем имена;

организуем массив (массивы) (при необходимости).

3. Выбираем метод решения задачи и разбиваем процесс ее решения на этапы с равным числом аналогичных операций.

Проверяем, есть ли для вычисления всех результатов формулы с фиксированным числом операций каждая.

Если есть, переходим в п. 6, в противном случае в п. 5.

5. Выводятся рабочие формулы, для чего:

а) записываются операции 1-го, 2-го, 3-го этапов решения задачи, используя выражения вида (1) и (2). Результат выполнения одной и той же операции на разных этапах обозначается одним именем с индексом, равным номеру этапа;

б) записываются операции, выполняемые на этапе с номером  $i$ , как результат обобщения операций п. 5а, т. е. выводятся рабочие формулы для вычисления всех переменных величин.

При этом каждая из этих величин выражается либо в виде функции номера этапа  $i$ , либо в виде функции предшествующего значений той же величины (итерационная формула);

в) записываются операции последнего этапа, если они известны;

г) все операции, выполняемые на каждом из рассмотренных этапов (обычно операции 1-го и последнего этапов), приводятся к виду формул  $i$ -го этапа.

При этом выявляются начальные значения переменных.

ба. Для каждой переменной величины, включенной в формулы  $i$ -го этапа, выписываем начальное, конечное значения и закон изменения. Для каждого массива вводим переменную, изображающую значения его индексов.

бб. Выявляем условие окончания (повторения) цикла. Оно может:

быть известно из условия задачи или метода ее решения;

определяться предельным значением некоторой переменной, полученным в п. ба, т. е. имеет вид отношения, связывающего текущее и предельное значения некоторой переменной;

быть получено, если известны операции последнего этапа: приравниваем выражения для некоторой переменной величины на  $i$ -м и последнем этапах.

7. Изображается типовая схема циклического алгоритма и заполняется каждый блок в соответствии с его назначением.

В частности, в рабочий блок записываются все выражения, полученные на этапе с номером  $i$  либо заданные рабочие формулы для вычисления результатов.

При заполнении схемы индексы сохраняются лишь для величин, рассматриваемых как массив.

Во всяком случае если значение величины используется только в пределах одного этапа, того же, на котором оно вычисляется (или следующего), то рассматривать величину как массив нет необходимости.

Если пункт 5б не может быть выполнен для всех величин, то это будет означать, что циклический алгоритм мы построить не сможем и нужно строить алгоритм другого вида либо следует представить алгоритм решения всей задачи как совокупность циклических и/или прочего вида алгоритмов, но в этом случае следует применять более общую методику (см. ниже).

42

### Примеры составления алгоритмов с одним циклом

В приводимых примерах мы не будем доказывать справедливость получаемых формул, поскольку этот процесс весьма схожий для всех задач.

#### Задача 1.

1. Вычислить значение функции  $Z = 3X^5 - X^4 + 6X^3 - 2X^2 - 7X + 3$  при одном значении  $X$ .

Решение.

2. Исходные данные:

- а)  $X$  (постоянная величина в пределах задачи);
- б) числа 3, -1, 6, -2, -7, 3 (коэффициенты многочлена).

Закон изменения последних отсутствует, следовательно, их обозначим как массив:  $A = \{a_0, a_1, \dots, a_5\} = \{3, -1, 6, -2, -7, 3\}$ .

Результат:  $Z$ .

5а. Порядок решения задачи.

1-й этап:  $b_1 = a_4X + a_5$ .

2-й этап:  $r_2 = XX$ ;  $b_2 = a_3r_2 + b_1$ .

3-й этап:  $r_3 = r_2X$ ;  $b_3 = a_2r_3 + b_2$ .

5б.  $i$ - этап:  $r_i = r_{i-1}X$ ;  $b_i = a_{5-i}r_i + b_{i-1}$

5в. Последний этап ( $k$ -й):  $r_k = r_{k-1}X$ ;  $b_k = a_0r_k + b_{k-1}$

Из сравнения индексов при  $a$  на  $i$ -м и  $k$ -м этапах вытекает условие повторения цикла:  $i < 5$ .

5г. Для единообразия 1-ю операцию 2-го этапа запишем так:  $r_2 = r_1X$ , т. е.  $r_1 = X$ . Операцию 1-го этапа дополним еще одной и запишем таким образом:

$$r_1 = r_0X; b_1 = a_4r_1 + b_0,$$

из чего следует:  $r_0 = 1$ ;  $b_0 = a_5$  (начальные значения).

6. Выделяем переменные:

$$\begin{array}{l} 1 \\ r | r := rX; \end{array} \quad \begin{array}{l} 1 \\ b | b := a_{5-i} + b; \end{array} \quad \begin{array}{l} 1 \\ i | i := i + 1. \\ 5 \end{array}$$

Индексы при переменных  $r, b$  опускаем.

7. Рисуем типовую схему алгоритма (см. рис. 1) и заполняем ее. Получаем рис. 3. В этой схеме и во всех последующих блоки ввода, вывода и останова условно не изображаем.

#### Задача 2.

1. Дана матрица  $A (n \times n)$ . Найти наибольший элемент побочной диагонали.

Решение.

2. Исходные данные:  $n$ , матрица  $A (n \times n)$ . Результат:  $b$ .

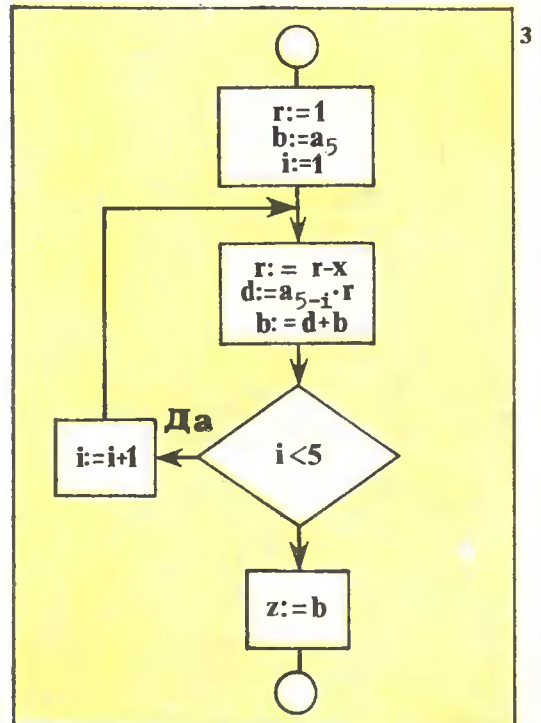
3. Метод: сравниваем первые два элемента диагонали и находим наибольший. Результат сравниваем с третьим элементом диагонали и находим наибольшее значение и т. д.

5а. Порядок решения задачи.

1-й этап: проверить условие  $a_{1,n} > a_{2,n-1}$ ; если условие выполняется, то наибольшее значение (обозначим его  $b_1$ ) равно  $a_{1,n}$  иначе  $b_1 = a_{2,n-1}$

2-й этап: проверить  $b_1 > a_{3,n-2}$ ; да —  $b_2 = b_1$ , нет —  $b_2 = a_{3,n-2}$

3-й этап: проверить  $b_2 > a_{4,n-3}$ ; да —  $b_3 = b_2$ , нет —  $b_3 = a_{4,n-3}$



5б.  $i$ -й этап: проверить  $b_{i-1} > a_{i+1, n-i}$ ; да —  $b_i = b_{i-1}$ , нет —  $b_i = a_{i+1, n-i}$

5в. Последний этап ( $r$ -й): проверить  $b_{r-1} > a_{n,1}$ ; да —  $b_r = b_{r-1}$ , нет —  $b_r = a_{n,1}$ .

Из сравнения индексов при  $a$  на  $i$ -м и  $r$ -м этапах вытекает условие ПЦ:  $i+1 < n$ , или  $i < n-1$ .

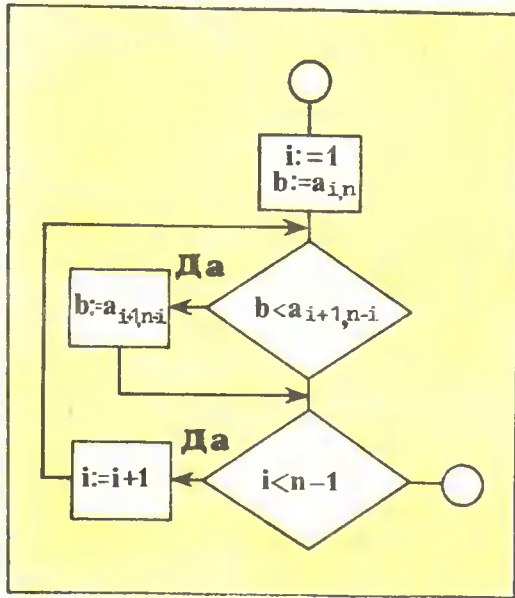
5г. Для единообразия операции 1-го этапа запишем так:  $b_a > a_{2, n-1}$ ; да —  $b_1 = b_0$ , нет —  $b_1 = a_{2, n-1}$ . Отсюда следует:  $b_0 = a_{1, n}$ .

6. Вывяляем переменные

$$i \begin{cases} i := i+1 \\ n-1 \end{cases} b \begin{cases} a_{1, n} \\ ? \end{cases} \begin{cases} b, & \text{если } b > a_{i+1, n-i} \\ a_{i+1, n-i}, & \text{если } b \leq a_{i+1, n-i} \end{cases}$$

Индекс при  $b$  опускаем, так как для вычисления любого последующего значения  $b$  достаточно знать лишь предшествующее значение.

7. Рисуем типовую схему циклического алгоритма и заполняем ее. Получаем окончательную схему (рис. 4).



Задача 3.

1. Вычислить приближенное значение функции  $Y = a^X$  ( $X < 0$ ) с помощью разложения ее в ряд:

$$Y = 1 + X \ln a / 1 + X^2 \ln^2 a / 2 + \dots$$

Допустимая погрешность —  $\epsilon$ .

В подобных задачах за приближенное значение функции принимается сумма  $n$  членов ряда. Погрешность  $\delta$  такого приближения принимается равной значению  $n$ -го члена ряда.

2. Исходные данные:  $a, X, \epsilon$ . Результат:  $Y$  (одно значение).

3. Метод: последовательное вычисление каждого члена ряда, начиная с первого, и сложение его с предшествующей суммой.

Обозначим  $c = \ln a$

5а. 1-й этап:  $\delta_1 = cX$ ;  $Y_1 = \delta_1 + 1$ .

2-й этап:  $r_2 = 1 \cdot 2$ ;  $\delta_2 = \delta_1 c X / r_2$ ;  $Y_2 = \delta_2 + Y_1$ .

3-й этап:  $r_3 = r_2 \cdot 3$ ;  $\delta_3 = \delta_2 c X / r_3$ ;  $Y_3 = \delta_3 + Y_2$ .

5б.  $i$ -й этап:  $r_i = r_{i-1} \cdot i$ ;  $\delta_i = \delta_{i-1} c X / r_i$ ;  $Y_i = \delta_i + Y_{i-1}$ .

Поскольку условие ОПЦ известно ( $|\delta| < \epsilon$ ), то операции последнего этапа не записываем.

5в. Для единообразия запишем операции 1-го и 2-го этапов так:

$r_2 = 2r_1$ ;  $r_1 = 1r_0$ ;  $\delta_1 = \delta_0 c X / r_1$ ;  $Y_1 = \delta_1 + Y_0$ .

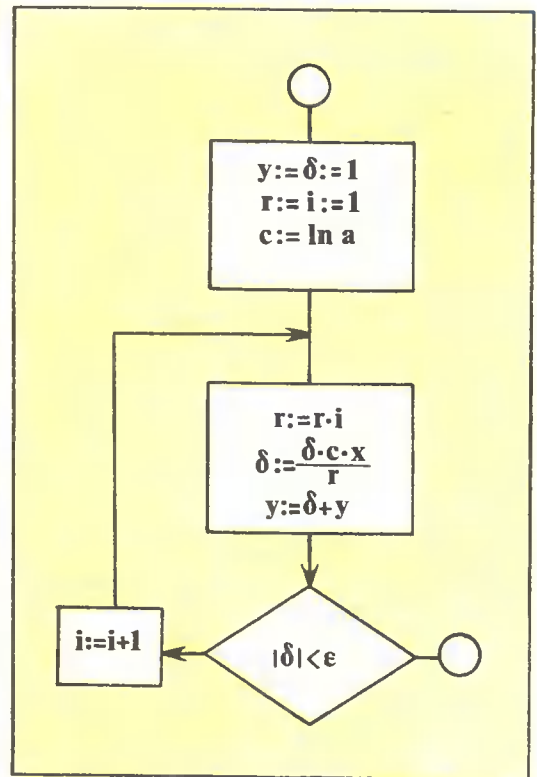
Отсюда следует, что начальные значения  $r, \delta$  и  $Y$  равны 1.

6.

$$i \begin{cases} i := i+1 \\ ? \end{cases} Y \begin{cases} Y := Y + \delta / r \\ ? \end{cases} r \begin{cases} r := r \cdot i \\ ? \end{cases}$$

$$\delta \begin{cases} \delta := \delta c X / r \\ \epsilon \end{cases}$$

7. Составляем схему (рис. 5).



Задача 4.

1. Из массива  $A(1:n)$  выбрать положительные элементы.

Решение.

2. Исходные данные:  $n, A(1:n)$ .

3. Результат: массив  $B(1:n)$ .

Пояснение. Число положительных элементов массива  $A$  заранее неизвестно, поэтому

«заказываем» массив В максимального размера.

4. Порядок решения задачи.

Сравниваем с нулем по порядку, начиная с 1-го, каждый элемент массива А; если его значение больше нуля, переносим его значение в массив В.

5. Порядок решения задачи по этапам:

Этап 1: проверить  $a_1 > 0$ ; да —  $b_1 = a_1$ , нет — переход к следующему этапу.

Этап 2: проверить  $a_2 > 0$ ; да —  $b_2 = a_2$ , нет — переход к следующему этапу.

Внимательный читатель уже заметил ошибку в этих формулах.

Действительно, индекс элемента массива В не совпадает с номером этапа. Так, на 2-м этапе он может принимать значения 1 или 2, на 3-м — 1, 2 или 3 и т. д., т. е. на каждом этапе значение индекса есть переменная величина, не зависящая от номера этапа. Поэтому и обозначим индекс именем, как переменную величину, например буквой  $f$ .

В соответствии с п. 3б методики для этой величины нужно вывести рекуррентную формулу, связывающую предшествующее и последующее значения этой величины. Эта формула очень проста и очевидна:

$$f = f + 1. \quad (7)$$

Очевидно также, что вычисление нового значения  $f$  (по формуле (7)) следует выполнять при каждом выявлении в массиве А очередного положительного элемента.

Согласно этому порядок решения задачи будет описан так.

Этап 1:  $a_1 > 0$ ; да —  $f = f + 1, b_f = a_1$ , нет — переход к следующему этапу.

Этап 2:  $a_2 > 0$ ; да —  $f = f + 1, b_f = a_2$ , нет — переход к следующему этапу.

Этап  $i$ :  $a_i > 0$ ; да —  $f = f + 1, b_f = a_i$ , нет — переход к следующему этапу.

Формулу (7) на первом этапе мы вписали ради единообразия; из нее вытекает, что начальное значение  $f = 0$ .

Теперь можно изобразить схему алгоритма (рис. 6).

Обобщая приведенные рассуждения, можно сказать, что в тех случаях, когда некоторая переменная зависит от ее предшествующего значения, мы выполняем три несложные операции:

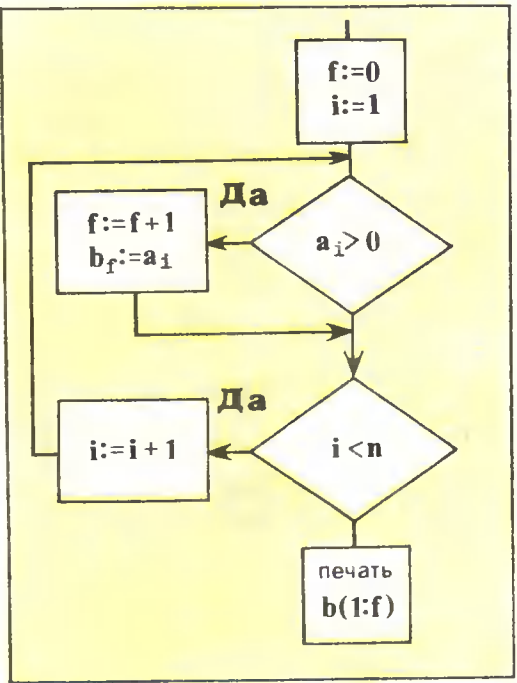
обозначаем ее именем (буквой  $f$ , например);

выводим рекуррентную формулу для этой величины ( $f = f + 1$ );

определяем положение этой формулы в описываемом процессе решения задачи.

Далее действуем в соответствии с методикой.

Этот же подход можно использовать и в том случае, если переменная зависит от номера этапа, но эта зависимость сложная.



Методика составления алгоритмов произвольного вида

Составляется укрупненная линейная, разветвляющаяся или циклическая (с одним циклом) схема алгоритма. Затем составляются схемы отдельных укрупненных блоков (независимо друг от друга). И наконец составляется подробная схема алгоритма решения всей задачи механическим объединением схем отдельных блоков в соответствии с укрупненной схемой.

При таком подходе процесс составления алгоритма предполагает несколько последовательных шагов, на каждом из которых составляется элементарная по своей структуре схема. Сложность всей задачи сказывается лишь на числе шагов.

1. Четко формулируется условие задачи.
2. Выделяются исходные данные и выбирается их структура, т. е. определяется, в виде какой совокупности массивов и переменных нужно представить данные задачи. Выделяются результаты (их количество, имена).
3. Формулируется метод решения задачи в наиболее общем виде (без лишних подробностей).
4. Выделяются наиболее крупные этапы решения задачи выполняющиеся однократно и охватывающие в совокупности процесс решения всей задачи (если такие этапы есть, то их не менее двух). Они последовательно описываются, со-

ставляется укрупненная линейная или разветвляющаяся схема алгоритма всей задачи с одним выходом и выполняется переход в п. 7.

При отсутствии указанных этапов перейти в п. 5.

5. Выделяется наиболее крупная операция, выполняющаяся многократно, т. е. операция, многократное решение которой обеспечивает решение всей задачи.

6. Составляется укрупненная схема алгоритма с одним циклом по известной методике. При этом, описывая процесс решения задачи в терминах операции п. 5, на каждом из этапов (1, 2, 3, ...,  $i$ ) указываем множества исходных данных и результатов задания, в общем случае, их граничных значений (или граничных значений индексов).

7. Для каждого укрупненного блока полученной схемы составляем свою схему алгоритма, рассматривая поочередно каждый блок как самостоятельную задачу и применяя к нему настоящую методику, начиная с п. 1.

Очередность рассмотрения блоков в одной укрупненной схеме — от выхода ко входу.

### Пример решения задач

#### Задача 5.

1. В матрице  $A(n, n)$  значения элементов, лежащих на главной диагонали и правее ее, сдвинуть на одну позицию вправо. Элементы последнего столбца запомнить, а элементы главной диагонали принять равными нулю.

#### Решение.

2. Исходные данные: матрица  $A(n, n)$ ,  $n$ . Результат: матрица  $A(n, n)$ , массив  $B = \{b_1, \dots, b_n\}$ ; элементы массива  $B$  должны быть равны элементам последнего столбца матрицы  $A$ .

3. Метод: последовательно преобразуем строки матрицы начиная с первой.

4. Анализируя порядок решения задачи, убеждаемся, что этот процесс распадается на ряд этапов, на каждом из которых выполняется аналогичный набор операций (обработка элементов одной строки). Однако обработка  $n$ -й строки отличается от обработки предшествующих строк, так как в  $n$ -й строке отсутствует сдвиг элементов.

Поэтому во всей задаче можно выделить две крупные однократно выполняемые операции, т. е. две подзадачи:

блок 1 (обработка строк с 1-й по  $(n-1)$ -ю);

блок 2 (обработка  $n$ -й строки).

5—6. Процесс решения задачи блока 1.

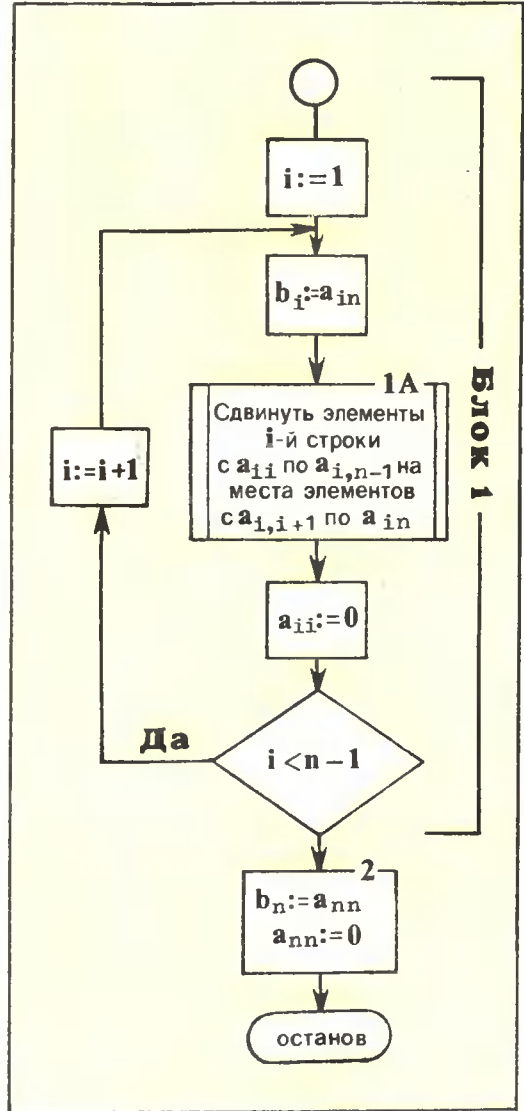
1-й этап:  $b_i = a_{i,n}$ ; сдвинуть элементы с  $a_{i,1}$  по  $a_{i,n-1}$  на места элементов с  $a_{i,2}$  по  $a_{i,n}$  соответственно;  $a_{i,1} = 0$ .

2-й этап:  $b_2 = a_{2,n}$ ; сдвинуть элементы

с  $a_{2,2}$  по  $a_{2,n-1}$  на места элементов с  $a_{2,3}$  по  $A_{2,n}$  соответственно;  $a_{2,2} = 0$ .

$i$ -й этап:  $b_i = a_{i,n}$ ; сдвинуть элементы с  $a_{i,1}$  по  $a_{i,n-1}$  на места элементов с  $a_{i,i+1}$  по  $a_{i,n}$  соответственно;  $a_{i,i} = 0$ .

На последнем этапе в данном случае обрабатывается  $(n-1)$ -я строка матрицы  $A$ , так что  $i = 1, n-1$ . Составляем укрупненную схему алгоритма решения всей задачи (рис. 7).



Составление схемы алгоритма задачи блока 1

1. Условие задачи: сдвинуть элементы матрицы  $A$  с  $a_{i,1}$  по  $a_{i,n-1}$  на одну позицию вправо.

Решение.

2. Исходные данные:  $a_{i,j}, \dots, a_{i,n-1}$ .

Результат:  $a_{i,i+1}, \dots, a_{i,n}$

3. Метод: последовательно переносим вправо каждый элемент начиная с последнего.

5. Наиболее крупная операция — перенос одного элемента — повторяется многократно.

6. Порядок решения задачи.

1-й этап:  $a_{i,n} := a_{i,n-1}$ .

2-й этап:  $a_{i,n-1} := a_{i,n-2}$ .

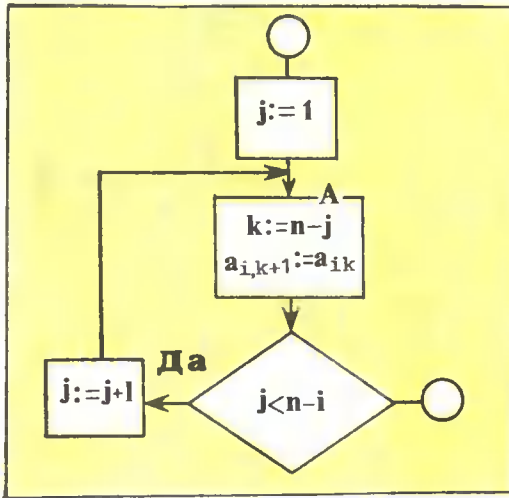
$j$ -й этап:  $a_{i,n-j+1} := a_{i,n-j}$ .

$r$ -й этап (последний):  $a_{i,i+1} := a_{i,i}$ .

Отсюда, сравнивая выражения для 2-го индекса результата операции на  $j$ -м и на  $r$ -м этапах, получаем условие ОПЦ:  $n-j+1 = i+1$  или  $j = n-i$ .

Составляем схему блока 1 (рис. 8).

8



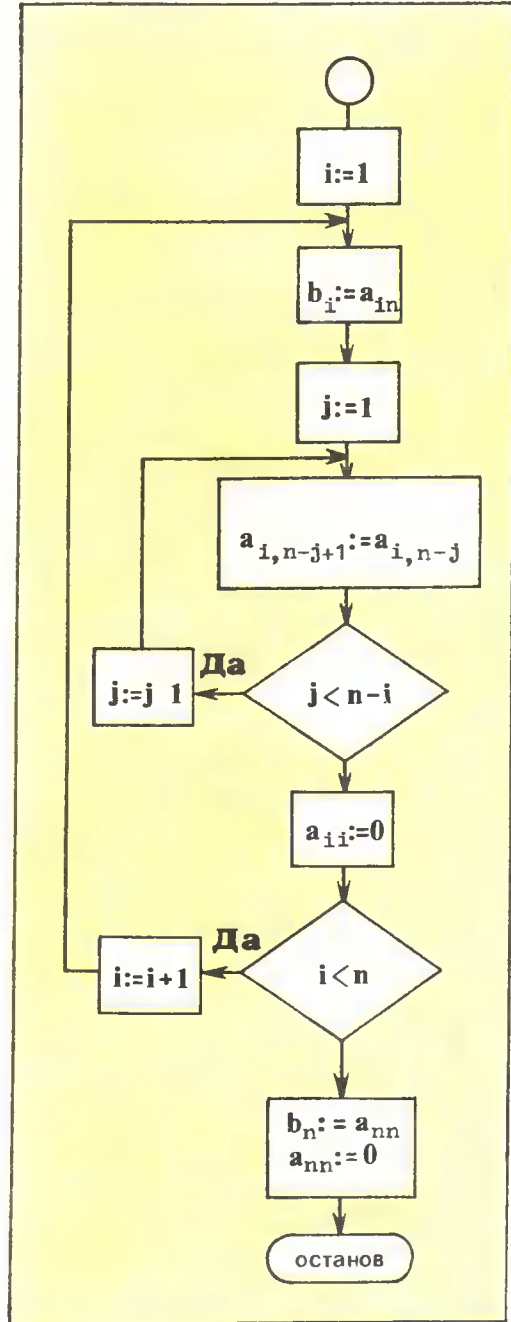
8. Составляем подробную схему для всей задачи (рис. 9).

### Закключение

Как уже отмечалось, изложенные методики носят эвристический характер, отсутствуют строгие определения многих понятий, формализация. В частности, отсутствует определение понятия «наиболее крупная операция». Здесь мы полагаемся на здравый смысл учащихся, и, как показывает опыт, это оправдано. Местами методики подвергались специальному упрощению с целью увеличения их доступности.

Однако и в таком виде эти методики способствуют быстрому овладению техникой составления алгоритмов. Они с 1974 г. используются в Ростовском инженерно-строительном институте в курсе «Вычислительная техника». Практическое применение показало их высокую эффективность. Опыт говорит, что 18—20 ч занятий достаточно для

9



того, чтобы научить основную массу учащихся составлению схем алгоритмов средней сложности. Они значительно дисциплинируют мышление при разработке алгоритма и позволяют сразу отсеять множество ненужных вариантов алгоритма, а искомый вариант получается не методом случайного подбора, как это делается обычно, а вы-



водится строго математически из условия задачи.

Достоинством методик является также и тот факт, что результатом их применения являются схемы алгоритмов с четкой, регулярной структурой, позволяющие легко выделять модули, удобные для внесения исправления, и т. д.

Кроме того, при обычном подходе преподаватель видит лишь конечный результат работы учащегося над составлением алгоритма — сам алгоритм. При применении методик указанный процесс становится наблюдаемым для преподавателя и, как побочный эффект, хорошо документированным.

## **Вниманию пользователей системы «РИГА-микро»!**

Кохтла-Ярвский городской центр НТТМ «Мыте» предлагает учебным заведениям, использующим в работе КУВТ-86, комплект дополнительной поставки системы «РИГА-микро», распространяемой центром на договорных основах (см.: Информатика и образование. 1988. № 6; 1989. № 1, 4).

### *Дополнительный комплект включает:*

- новую версию управляющей программы системы «РИГА» с существенно расширенными функциями, в их числе: возможность получения экранных копий (с рисунками), сохранение их на дискете и распечатка в графическом виде; возможность совмещения рисунков, сохраненных на дискете и пересылаемых на БК, с диалогом, организуемым преподавателем; возможность распечатки любых текстов со сменными алфавитами; повышена общая надежность сети, внесены дополнительные возможности в режимы «Прием из БК», «Печать», «Электронная почта»;

- систему подготовки текстов ТЕКСТ1, предназначенную для подготовки и редактирования текстов сложной структуры на БК-0010; имеется возможность использования при печати помимо стандартного целого ряда графических алфавитов, специально для этого созданных и загружаемых по мере необходимости; возможна работа на отдельных БК или в составе КУВТ-86, с печатающими устройствами ROBOTRON или D-100;

- графический редактор RIS, предназначенный для создания на экране дисплея несложных рисунков, блок-схем, геометрических объектов; работа ведется с использованием графических шаблонов, таких, как точки, линии, круги, конусы, кубы и т. п.;

- пакет обучающе-контролирующих программ по английскому языку (13 программных файлов), предназначенных для организации повторения, систематизации, закрепления и контроля знаний учащихся по ряду основных тем учебной программы по английскому языку;

- пакет демонстрационно-обучающих программ по физике (25 программных файлов), охватывающих темы «Механика», «Электричество», «Ядерная физика»; используются при повторении пройденного материала и в самостоятельной работе учащихся для закрепления полученных знаний;

- дискету новых динамических компьютерных игр.

Прилагаются необходимые документация и инструкции по использованию нового программного обеспечения.

Стоимость полного комплекта (6 дискет) — договорная.

Реквизиты для заключения договора (почтовый адрес, Ф. И. О. руководителя, телефон, расчетный счет организации, МФО банка) просьба направлять по адресу:

**202020. Кохтла-Ярве, ул. Циолковского, 7—21.**

**Директору центра НТТМ «Мыте» Путконен М. А.  
Телефон для справок: 26-510 (код Кохтла-Ярве: 01433).**

## Решение задач с помощью Бейсика

В Центре информатики и профориентации г. Троицка создано пособие по программированию на Бейсике — сборник задач по разным предметам школьного курса, предназначенный для старшеклассников, освоивших основы этого языка. Предлагаем вниманию наших читателей отдельные задачи из пособия.

Как правило, написанию программы на каком-либо языке предшествует разработка алгоритма. Алгоритм можно записывать на некоем псевдоязыке (псевдокоде) или на подходящем реальном языке программирования (например, на Паскале). Предполагается, что язык, используемый в таких целях, содержит основные управляющие конструкции структурного программирования и допускает вызов процедур. Это позволяет реализовать структурный подход к разработке алгоритма. В настоящее время такой подход считается предпочтительным, по крайней мере на этапе обучения программированию.

Ниже при разработке алгоритмов будут использоваться следующие управляющие конструкции:

Простое следование операторов (операторы разделяются точкой с запятой).  
если... то... конец если  
если... то... иначе... конец если  
от... до... шаг... выполнять...  
конец цикла  
до тех пор пока... выполнять... конец пока  
повторять... пока не...

Служебные слова управляющих конструкций выделяются жирным шрифтом.

Поскольку Бейсик не имеет операторов, в точности реализующих вышеприведенные управляющие конструкции, то при их реали-

зации приходится использовать наборы операторов, и в частности оператор безусловного перехода GOTO. Для сохранения структурированности программ предлагается осуществлять перевод с алгоритмического языка на Бейсик по возможности стандартным способом. Ниже в виде примеров приводится один из возможных способов такого перевода.

1. Фрагмент программы, получающий число с терминала и добавляющий его к числу СУММА, если оно положительное.

Алгоритм

ввод X;

если  $X > 0$  то сумма = сумма + X конец если

Реализация на Бейсике

```
10 INPUT X
20 IF X<=0 GOTO 40
30 S=S+X
40 REM Продолжение программы
```

Если должно выполняться только одно действие, то Бейсик допускает более простую реализацию:

```
20 IF X>0 THEN S = S + X
```

Первый способ более общий, так как позволяет записать любое количество операторов после выполнения условия.

2. Фрагмент программы, получающий два числа и печатающий их в порядке возрастания.

Алгоритм

ввод X, Y;

если  $X > Y$  то печатать X, Y

иначе печатать Y, X

конец если

Реализация на Бейсике

```
10 INPUT X,Y
20 IF X<Y THEN GOTO 30
30 PRINT X,Y
```

```

40 GOTO 60
50 PRINT Y, X
60 REM Продолжение программы

```

3. Фрагмент программы, получающий целое число и вычисляющий его факториал.

Алгоритм  
ввод N;  
 $F = 1; K = 0;$   
до тех пор пока  $K <> N$  выполнять  
 $K = K + 1; F = F * K$

конец пока  
Реализация на Бейсике

```

10 INPUT N%
20 F%=1
30 K%=0
40 IF K%=N% GOTO 80
50 K%=K%+1
60 F%=F%*K%
70 GOTO 40
80 REM Продолжение программы

```

4. Та же операция может быть реализована с помощью другой управляющей конструкции.

Алгоритм  
ввод N  
 $F = 1; K = 0;$   
повторять  
 $K = K + 1; F = F * K;$   
пока не  $K = N;$   
Реализация на Бейсике

```

10 INPUT N%
20 F%=1
30 K%=0
40 K%=K%+1
50 F%=F%*K%
60 IF K%>N% GOTO 40
70 REM Продолжение программы

```

5. Фрагмент, печатающий таблицу синусов для ряда чисел.

Алгоритм  
от  $X = 0$  до  $X = \pi$  шаг 0.1 выполнять  
 $Y = \sin(X);$  печатать X, Y  
конец цикла

Эта конструкция реализуется на Бейсике непосредственно.

```

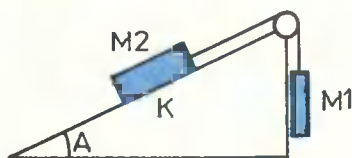
10 FOR X=0 TO PI STEP .1
20 Y=SIN(X)
30 PRINT X,Y
40 NEXT X

```

6. Вызов подпрограммы. Поскольку в Бейсике при вызове подпрограммы передача параметров не осуществляется, т. е. фактически понятие процедуры не реализовано, то вызов подпрограммы оператором GOSUB имеет смысл только в случае многократного использования одного и того же фрагмента с возвратами в различные точки программы.

**Задача 1. Движение по наклонной плоскости.**

Рассматривается ситуация, изображенная на рис. 1. Задаются массы  $M_1$  и  $M_2$ , коэффициент трения  $K$ , угол наклона плоскости  $A$ .



Требуется выяснить, движется груз  $M_1$  вверх, вниз или вся система неподвижна.

Рассмотрим силы, действующие на груз  $M_2$ . Это проекция силы тяжести  $M_2 \cdot g \cdot \sin A$ , сила натяжения нити  $M_1 \cdot g$  и если груз движется, то сила трения  $M_2 \cdot g \cdot \cos A \cdot K$ , направленная против направления движения.

Используя эти соотношения, напомним алгоритм ответа на поставленный вопрос.

начало  
ввод  $M_1, M_2, K, A;$   
если  $M_1 > M_2 * (\sin(A) + K * \cos(A))$  то движется вниз иначе  
если  $M_1 < M_2 * (\sin(A) - K * \cos(A))$  то движется вверх иначе неподвижен  
конец если

```

конец если
конец
10 REM - ДВИЖЕНИЕ ПО НАКЛОННОЙ
20 REM - ПЛОСКОСТИ
30 REM - ЧИСЛЕННЫЙ ЭКСПЕРИМЕНТ
40 PRINT "ВВЕДИТЕ M1, M2, K, A"
50 INPUT M1, M2, K, A
60 LPRINT "M1="; M1; "M2="; M2; "K="; K;
    "A="; A
70 A=AX3.14159/180
80 IF M1>M2*(SIN(A)+K*COS(A)) THEN
    LPRINT "ВНИЗ"; GOTO 110
90 IF M2*(SIN(A)-K*COS(A))>M1 THEN
    LPRINT "ВВЕРХ"; GOTO 110
100 LPRINT "НЕПОДВИЖЕН"
110 END

```

Пример работы программы

```

M1= 1 M2= 2 K= .3 A= 10
ВНИЗ
M1= 1 M2= 2 K= .5 A= 70
ВВЕРХ
M1= 1 M2= 10 K= .3 A= 45
ВВЕРХ

```

**Задача 2. Принцип Ферма.**

В школьном курсе физики изучается закон преломления световых лучей, который гласит, что отношение синуса угла падения к синусу угла преломления равно коэффициенту преломления на границе двух сред.

Этот закон является следствием более общего вариационного принципа, который называется принципом Ферма (который в свою очередь есть следствие общезначимого принципа минимального действия). Он состоит в следующем: луч света распространяется между двумя точками по такому пути, время прохождения которого минимально.

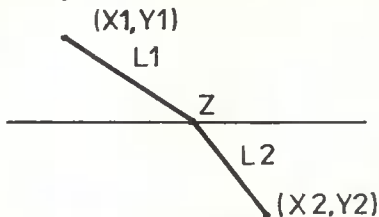
Взяв две точки по разные стороны границы

раздела двух сред, записав время прохождения света из одной точки в другую через любую точку границы раздела, взяв производную по координате этой точки и приравняв ее к нулю, мы найдем из принципа Ферма истинную точку пересечения лучом света границы двух сред.

Продолжаем эту операцию численно, для чего надо выполнить следующие операции.

1. Ввести координаты точки, из которой выходит луч:  $(X_1, Y_1)$  (рис. 2).

2



2. Ввести координаты точки, в которую луч приходит:  $(X_2, Y_2)$ .

50

3. Ввести требуемую точность  $N$  нахождения точки пересечения лучом границы двух сред (границей является плоскость  $Y=0$ ).

4. Простым перебором точек при  $Y=0$  (разумными пределами перебора будут  $X_1$  и  $X_2$ ) найти  $X$ , для которого время прохождения лучом пути из  $X_1$  в  $X_2$  будет минимально.

5. Вычислить отношение синуса угла падения к синусу угла преломления и сравнить с заданным коэффициентом преломления.

Будем полагать, что скорость распространения света в верхней среде равна  $C$ , а в нижней —  $C/N$ . Тогда из рис. 2 видно, что время распространения света есть

$$T = \frac{L_1}{C} + \frac{L_2 \cdot N}{C}$$

Поскольку нас интересует только значение  $X$ , при котором эта величина минимальна, то деление на константу  $C$  можно не производить.

начало

ввод  $X_1, X_2, Y_1, Y_2, N, N$ ;

$T_1$  = число, заведомо большее, чем  $T_{MIN}$ ;  
от  $X = X_1$  до  $X_2$  шаг  $N$  выполнять

вычислить  $L_1$ ;

вычислить  $L_2$ ;

$T = L_1 + N * L_2$ ;

если  $T < T_1$  то

$T_1 = T$ ;  $Z = X$

конец если

конец цикла

вывод

конец

10 REM ПРИНЦИП ФЕРМА

20 PRINT "ГРАНИЦА СРЕД: Y=0"

30 PRINT "ВВЕДИТЕ X1, X2, Y1>0, Y2<0"

40 INPUT X1, X2, Y1, Y2

50 PRINT "ВВЕДИТЕ КОЭФФИЦИЕНТ

ПРЕЛОМЛЕНИЯ"

60 INPUT N

70 PRINT "ВВЕДИТЕ ШАГ ПО X"

80 INPUT H

90 T1=N\*SQR((X2-X1)^2+(Y2-Y1)^2)

100 FOR X=X1 TO X2 STEP H

110 L1=SQR((X-X1)^2+Y1^2)

120 L2=SQR((X-X2)^2+Y2^2)

130 T=L1+L2\*N

140 IF T<T1 THEN T1=T:Z=X

150 NEXT X

160 X=Z

170 L1=SQR((X-X1)^2+Y1^2)

180 L2=SQR((X-X2)^2+Y2^2)

190 N1=(ABS(X1-X)/L1)/(ABS(X-X2)/L2)

200 PRINT "N1=",N1

210 LPRINT "ЗАДАННЫЙ КОЭФФИЦИЕНТ  
ПРЕЛОМЛЕНИЯ -";N

220 LPRINT "X1=";X1,"Y1=";Y1

230 LPRINT "X2=";X2,"Y2=";Y2

240 LPRINT "ШАГ ПО X -";H

250 LPRINT "ТОЧКА ПЕРЕСЕЧЕНИЯ ГРАНИЦЫ  
СРЕД X=";X

260 LPRINT "ОТНОШЕНИЕ СИНУСОВ ";N1

270 END

В программе в строках 180—200 дополнительно вычисляется отношение синуса угла падения к синусу угла преломления и сравнивается с вводимым значением коэффициента преломления.

Пример работы программы

ЗАДАННЫЙ КОЭФФИЦИЕНТ ПРЕЛОМЛЕНИЯ - 1.45

X1= 0 Y1= 3

X2= 10 Y2=-7

ШАГ ПО X - .05

ТОЧКА ПЕРЕСЕЧЕНИЯ ГРАНИЦЫ СРЕД X= 4.9

ОТНОШЕНИЕ СИНУСОВ 1.44831398575

Задача 3. Моделирование броуновского движения.

При броуновском движении частицы совершают случайные перемещения как по направлению, так и по длине. При этом они все дальше удаляются от своего первоначального положения: хотя очевидно, что средние координаты большого числа частиц должны оставаться постоянными (статистически), но среднее расстояние от каждой частицы до ее первоначального положения (модуль разности координат) растет. Например, рассмотрев одномерное движение частицы с шагом, постоянным по длине (равным 1), но случайным по направлению (+ или -), можно вычислить это среднее смещение относительно первоначального положения. Пусть после  $N$ -го шага расстояние от начального положения до частицы (положительное число) равно  $S_n$ , тогда с вероятностью 0,5 после  $N+1$ -го шага это расстояние будет  $S_n+1$  или  $S_n-1$ . Вычислим среднее значение расстояния  $S_{n+1}$ . Для этого найдем средний квадрат расстояния:

$$\bar{S}_{n+1}^2 = \frac{(S_n+1)^2 + (S_n-1)^2}{2} = \bar{S}_n^2 + 1.$$

Теперь, учтя, что  $S_1=1$ , получим

$$\bar{S}_{n+1}^2 = N+1 \text{ и } \bar{S}_n = N$$

для любого N.

Таким образом, для единичного шага среднее расстояние от первоначального положения равно корню из числа шагов. Ниже приведена программа, проверяющая, выполняется ли это соотношение для большого числа частиц при двухмерном движении, когда направление каждого шага выбирается случайно, а длина шага случайным образом меняется от 0,5 до 1,5 (среднее значение равно 1).

Будем считать, что теоретическое значение верно, если оно попадает в интервал (экспериментальное значение минус дисперсия, экспериментальное значение плюс дисперсия).

Дисперсию вычисляем по формуле

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (S_i - \bar{S})^2}{N-1}}$$

Здесь N — количество частиц, участвующих в испытании.

```

10 REM БРОУНОВСКОЕ ДВИЖЕНИЕ
20 DIM S(100)
30 PRINT "ВВЕДИТЕ ЧИСЛО ЧАСТИЦ < 100"
40 INPUT N
50 PRINT "ВВЕДИТЕ ЧИСЛО ШАГОВ"
60 INPUT M
70 S2=0
80 FOR I=1 TO N
90 X=0
100 Y=0
110 FOR J=1 TO M
120 L=.5+RND(1)
130 A=2*3.1416*RND(1)
140 X=X+L*SIN(A)
150 Y=Y+L*COS(A)
160 NEXT J
170 S(I)=SQR(X*X+Y*Y)
180 S2=S2+S(I)
190 NEXT I
200 S1=S2/N:REM СРЕДНЕЕ РАССТОЯНИЕ
210 D=0
220 FOR I=1 TO N
230 D=D+(S(I)-S1)^2
240 NEXT I
250 D1=SQR(D/(N-1)):REM ДИСПЕРСИЯ
260 LPRINT "КОЛИЧЕСТВО ЧАСТИЦ ";N
270 LPRINT "ТЕОРЕТИЧЕСКОЕ"
275 LPRINT "СРЕДНЕЕ УДАЛЕНИЕ ";SQR(M)
280 LPRINT "РАСЧЕТНОЕ"
285 LPRINT "СРЕДНЕЕ УДАЛЕНИЕ ";S1
290 LPRINT "ДИСПЕРСИЯ ";D1
300 END

```

Пример работы программы

```

КОЛИЧЕСТВО ЧАСТИЦ 10
ТЕОРЕТИЧЕСКОЕ
СРЕДНЕЕ УДАЛЕНИЕ 6.324555320337
РАСЧЕТНОЕ
СРЕДНЕЕ УДАЛЕНИЕ 5.7666837511725
ДИСПЕРСИЯ 3.2363558400885

```

Задача 4. Вычисление длины кривой.

Вычислить длину кривой, являющейся графиком функции SIN(X) на интервале (0; 3,14).

Вычисление длины кривой можно свести к суммированию отрезков кривой, которые на каждом сегменте разбиения заменяются отрезками прямой. Длина такого отрезка вычисляется по теореме Пифагора. Например, для X1, X2:

$$L = \text{SQR}((X2-X1)^2 + (F(X1)-F(X2))^2)$$

Точность вычислений определяется числом сегментов разбиения.

```

10 REM ВЫЧИСЛЕНИЕ ДЛИНЫ КРИВОЙ
20 DEF FNF(X)=SIN(X):REM СЮДА
   ВСТАВЛЯЕТСЯ ЛЮБАЯ ФУНКЦИЯ
30 PRINT "ВВЕДИТЕ XMIN, XMAX, ЧИСЛО
   ТОЧЕК РАЗБИЕНИЯ"
40 INPUT X1,X2,N
50 H=(X2-X1)/N
60 L=0
70 FOR I=1 TO N
80 L=L+SQR((FNF(X1+H*I)
   -FNF(X1+H*(I-1)))^2+H^2)
90 NEXT I
100 LPRINT "XMIN=";X1,"XMAX=";X2,
   "N=";N:LPRINT "L=";L
110 END

```

Пример работы программы

```

XMIN= 0           XMAX= 3.14       N= 10
L= 3.813035329177

XMIN= 0           XMAX= 3.14       N= 100
L= 3.8178962168057

```

Задача 5. Метод Монте-Карло.

Найдем площадь под графиком функции, например SIN(X) на интервале (0, π), несколько необычным методом. Пусть график нарисован на прямоугольнике 1×π, площадь его — π. Предположим, что мы стреляем по этому прямоугольнику и каждый раз попадаем в случайную точку внутри него. Тогда площадь можно оценить из соотношения

π · (количество попаданий под кривую) / (общее число попаданий).

Очевидно, что точность метода, реализация которого на ЭВМ требует генерации случайных чисел, зависит от количества этих чисел, качества генератора и, как правило, невелика по сравнению с другими численными методами, хотя программа работает долго.

Поэтому кажется, что такой способ нахождения площади не очень удачен (и это действительно так), однако существуют задачи, которые поддаются решению только методом Монте-Карло, и в этом смысле он может претендовать на место самого универсального численного метода. Так что познакомиться с ним следует.

```

10 REM НАХОЖДЕНИЕ ПЛОЩАДИ
   МЕТОДОМ МОНТЕ-КАРЛО
20 DEF FNF(X)=SIN(X)

```

```

30 PI=3.1416
40 PRINT "ВВЕДИТЕ ДЛИНУ
ПОСЛЕДОВАТЕЛЬНОСТИ СЛУЧАЙНЫХ ЧИСЕЛ"
50 INPUT N
60 M=0
70 FOR I=1 TO N
80 X=PI*RND(1)
90 Y=RND(1)
100 IF FNF(X)>Y THEN M=M+1
110 NEXT I
120 LPRINT "СГЕНЕРИРОВАНО СЛУЧАЙНЫХ
ЧИСЕЛ: ";N
130 LPRINT "ПЛОЩАДЬ РАВНА"; (M/N)*PI
140 LPRINT "ТОЧНОЕ ЗНАЧЕНИЕ
ПЛОЩАДИ - 2.0"
150 END

```

### Пример работы программы

СГЕНЕРИРОВАНО СЛУЧАЙНЫХ ЧИСЕЛ: 10  
ПЛОЩАДЬ РАВНА 2.51328  
ТОЧНОЕ ЗНАЧЕНИЕ ПЛОЩАДИ - 2.0

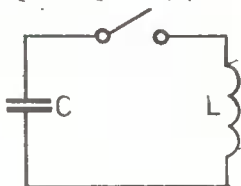
СГЕНЕРИРОВАНО СЛУЧАЙНЫХ ЧИСЕЛ: 100  
ПЛОЩАДЬ РАВНА 1.88496  
ТОЧНОЕ ЗНАЧЕНИЕ ПЛОЩАДИ - 2.0

### Задача 6. LC-контур.

Конденсатор колебательного контура, изображенного на рис. 3, заряжен до заряда  $Q$ . В начальный момент времени ключ замыкается, и начинаются колебания. Найти численно, исходя из основных уравнений, как зависит заряд на обкладках конденсатора от времени для нескольких первых циклов колебаний. Результат сравнить с точным решением

$$Q(t) = Q_0 \cos(t/\sqrt{LC}).$$

3



После замыкания ключа в цепи потечет ток  $I = \frac{\Delta Q}{\Delta t}$ , где  $Q$  — мгновенное значение заряда на обкладках (которое нам надо найти). Этот ток вызывает ЭДС индукции, равную  $-L \frac{\Delta I}{\Delta t}$ . Поскольку никаких других элементов в цепи нет, эта ЭДС равна потенциалу на обкладках конденсатора:

$$Q/C = -L(\Delta(\Delta Q/\Delta t))/\Delta t.$$

Мы получили так называемое уравнение в конечных разностях, которое нам и предстоит решить. Чтобы не исследовать предварительно вопрос о требуемой точности, шаг по времени  $T$  и общее количество шагов  $N$  будем вводить в режиме диалога.

Метод численного решения такого уравнения прост. Мы считаем, что в каждый следующий момент времени  $Q = Q + \Delta Q$  и  $\Delta Q = \Delta Q + \Delta(\Delta Q)$ .

Обозначим еще две переменные, которые

нам потребуются:  $Q1 - \Delta Q$ ,  $Q2 - \Delta Q1$ .  $\Delta(\Delta Q)$  находится из конечно-разностного уравнения, в котором в качестве  $Q$  берется предыдущее значение.

начало  
ввод  $L, C, T, N, Q0$ ;  
 $Q = Q0$ ;

от  $I=1$  до  $N$  шаг 1 выполнить

$$Q2 = (-Q * I) / (L * C);$$

$$Q1 = Q1 + Q2;$$

$$Q = Q + Q1;$$

вывод  $Q$ , точного  $Q$  (вычисленного по формуле)

конец цикла

конец

При реализации этого алгоритма нужно учесть, что для достижения хорошей точности шаг  $T$  должен быть, по всей видимости, достаточно малым, и, чтобы увидеть на экране терминала хотя бы один или два полных цикла колебаний, мы должны выводить не каждое полученное значение  $Q$ , а, например, каждое  $M$ -е, где  $M$  — число, дополнительно вводимое нами. В программе этот вывод реализован в строках 120—130.

```

10 REM LC-КОНТУР
20 PRINT "ВВЕДИТЕ L,C,T,Q0,N,M"
30 INPUT L,C,T,Q0,N,M
40 Q=Q0
50 LPRINT "L=";L;"C=";C;"Q0=";Q0
60 LPRINT " T Q Q(ТОЧНОЕ)"
70 FOR I=1 TO N
80 Q2=(-Q*T*T)/(L*C)
90 Q1=Q1+Q2
100 Q=Q+Q1
110 IF M*INT(I/M)<>I GOTO 130
120 LPRINT T*I;Q;
Q0*XCOS((I/SQR(L*C))*T*I)
130 NEXT I
140 END

```

### Пример работы программы

L=1	C=1	Q0=100	Q(ТОЧНОЕ)
.1	99.450494828429	99.500416527803	
.2	97.907304912649	98.006657784126	
.3	95.385849424576	95.53364891256	
.4	91.911322123738	92.106099400289	
.5	87.518439627464	87.758256189037	
.6	82.251094531029	82.533561490967	
.7	76.161916843708	76.484218728447	
.8	69.311748122729	69.670670934715	
.9	61.769033559491	62.160996827065	
1	53.609138092141	54.030230586811	

### Задача 7. Лисы и зайцы.

На острове проживает сообщество лис и зайцев. Известно начальное количество тех и других. В отсутствие лис зайцы размножаются с постоянной скоростью (предполагаем, что Мальтус был прав), а в отсутствие зайцев лисы (которым становится нечего есть) вымирают с постоянной скоростью. Скорость размножения лис линейно зависит от количества пищи (зайцев). Как будет меняться со временем количество

лис и зайцев на острове? Рассмотреть случай неограниченного количества пищи для зайцев (травы) и ограниченного, когда вводится скорость вымирания зайцев, растущая с ростом количества зайцев.

Обозначим число зайцев  $N$ , число лис  $P$ . Если есть только зайцы и пищи у них достаточно, то для темпа роста численности популяции применим закон Мальтуса  $\Delta N/\Delta t = r_n N$ . Если есть и лисы, то они поедают зайцев со скоростью, зависящей от числа лис. Тогда простейшая линейная модель динамики популяции зайцев даст следующее конечно-разностное уравнение:

$$\Delta N/\Delta t = (r_n - C_1 P) N.$$

Пусть  $r_p$  — скорость вымирания лис в отсутствие зайцев. Скорость размножения лис растет с ростом числа зайцев, линейное приближение дает

$$\Delta P/\Delta t = (-r_p + C_2 N) P.$$

Если теперь учесть ограниченность ресурсов для зайцев, то необходимо ввести коэффициент  $B$ , который, будучи умноженным на текущее количество зайцев, даст скорость их вымирания. Система запишется в виде:

$$\begin{cases} \Delta N/\Delta t = (r_n - C_1 P - BN) N, \\ \Delta P/\Delta t = (-r_p + C_2 N) P. \end{cases}$$

Алгоритм решения задачи несложен и сводится к решению системы двух конечно-разностных уравнений, способ решения полностью аналогичен решению одного уравнения (см. задачу 6). Решение реализовано в строках 320—350. Оно наглядно представляется в графическом виде (как, впрочем, и у многих других задач). Поэтому вся остальная часть программы — псевдографический вывод на принтер (полностью аналогичный выводу на терминал).

10 REM ЛИСЫ И ЗАЙЦЫ: УРАВНЕНИЕ  
ЛОТКИ-ВОЛЬТЕРРА

```

20 DIM A$(40,22)
30 FOR J=0 TO 22
40 FOR I=0 TO 40
50 A$(I,J)=""
60 NEXT I
70 NEXT J
80 PRINT "ВВЕДИТЕ КОЛИЧЕСТВО ЛИС
И ЗАЙЦЕВ"
90 INPUT N2,N1
100 PRINT "ВВЕДИТЕ R1,R2,C1,C2,B"
110 INPUT R1,R2,C1,C2,B
120 PRINT "ВВЕДИТЕ ОБЩЕЕ ВРЕМЯ
И ПЕРИОД ВЫВОДА"
130 INPUT T1,T2
140 X1=0:X2=T1:T=1
150 Y1=0:Y2=110:YX=0
160 FOR I=0 TO T1
170 IF T2<INT(I/T2)<>I GOTO 280
180 YX=INT((N1/110)*X2)
190 IF YX>22 THEN YX=22
200 IF YX<0 THEN YX=0
210 A$(X,Y)=""
220 YX=INT((N2/110)*X2)
230 IF YX>22 THEN YX=22

```

```

240 IF YX<0 THEN YX=0
250 A$(X,Y)=""
260 X=X+1
270 PRINT I;N1;N2
280 D1=(R1-C1*N2-B*N1)*N1
290 D2=(-R2+C2*N1)*N2
300 N1=N1+D1
310 N2=N2+D2
320 NEXT I
390 INPUT L$
400 IF L$<>"P" GOTO 30
410 OPEN "LPT:" FOR OUTPUT AS #1
420 PRINT #1,"R1=";R1,"R2=";R2
430 PRINT #1,"C1=";C1,"C2=";C2
440 PRINT #1,"B=";B
450 PRINT #1," "
460 FOR J=22 TO 0 STEP -1
470 FOR I=0 TO 40
480 PRINT #1,A$(I,J);
490 NEXT I
500 PRINT #1," "
510 NEXT J
520 CLOSE #1
530 GOTO 30
540 END

```

Пример работы программы

```

R1=2 .02          R2= .03
C1= 1E-04        C2= 3E-04
B= 0

```

```

ЗЗЗЗЗЛЛЛЛЛЛЛЛЛЛЛЛЛЛЛЛЛЛЛЛЛ      ЗЗЗЗЗЗЛЛЛ
                                     Л
Э           Л                       Л           Э
                                     Л           Л           Э
                                     Л           Л           Л           Л
Л           Э                       Л           Л           Л           Л
ЛЛ          Э                       ЭЛ          ЛЛ          Л
Л           Э                       Э          ЛЛ          Л
                                     Э           Э
ЗЗЗЗЗ

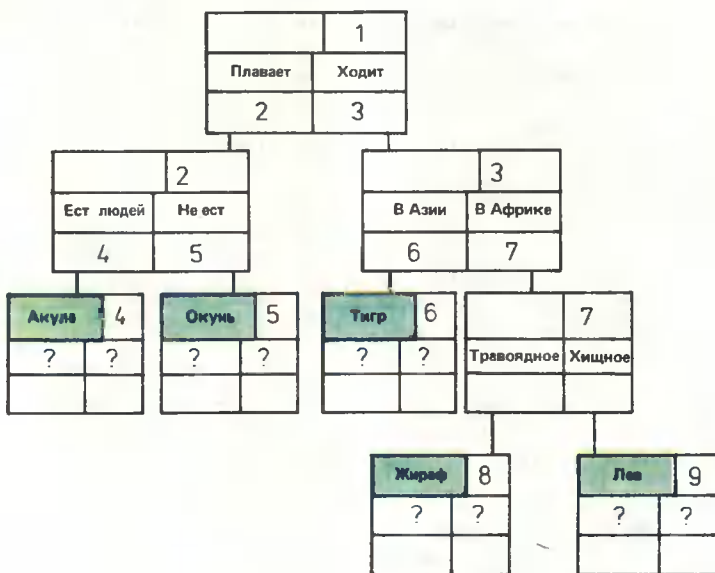
```

Задача 8. Угадай зверя.

Данная программа строит бинарное дерево, которое позволяет найти требуемый объект по его отличительным признакам. Если необходимый объект отсутствует, то можно ввести его название вместе с отличительным признаком. Первоначально дерево состоит из одной вершины. Для реализации наращивания дерева желательно динамическое распределение памяти, но, поскольку в Бейсике такая возможность отсутствует, мы заранее будем отводить память под вновь появляющиеся вершины. Для описания дерева используем «массив записей», реализованный в виде пяти независимых массивов, т. е. каждая вершина дерева описывается записью из пяти переменных:

$A(1,M)$  — имя узла;

$A(2,M)$  — отличительный признак левого объекта (признак1);



54

A(3,M) — отличительный признак правого объекта (признак2) (если вершина дерева является терминальной, то последние две переменные равны «?»);

V(1,M) — номер записи, в которой хранится левая дочерняя вершина (ссылка1);

V(2,M) — номер записи, в которой хранится правая дочерняя вершина (ссылка2);

M — номер записи, в которой хранится данная вершина.

ИМЯ(M)	
ПРИЗНАК 1(M)	ПРИЗНАК 2(M)
ССЫЛКА 1(M)	ССЫЛКА 2(M)

Для примера на рис. 4 приведено дерево, полученное в результате работы, протокол которой приведен в конце описания задачи.

Из рисунка видно, что фактически каждая вершина содержит либо имя, либо ссылки и признаки.

Алгоритм программы предусматривает спуск по дереву до терминальной вершины и при необходимости создание дочерних вершин.

начало

M=1; N=1;

если признак1(M) = «?» то обработка терминальной вершины  
иначе спуск по дереву

конец если

конец

Обработка терминальной вершины  
вывод имя(M);

если угадал то конец работы  
иначе

ввод имя; (\* новое имя \*)

ввод признак1;

ввод признак2;

замена признаков текущей записи;

создание двух дочерних записей

конец если;

Спуск по дереву

до тех пор пока признак1(M) <> «?»

запрос одного из двух отличительных признаков;

переход к записи по выработанному признаку;

конец пока;

Возможные усовершенствования: данная программа чисто демонстрационная, для реальной игровой программы необходимо увеличить размерность всех массивов с 20 до 100—200 и ввести запись последнего состояния дерева в файл, так как в данном варианте при каждом запуске программы надо начинать сначала.

10 REM БИНАРНОЕ ДЕРЕВО

20 DIM A\$(3,20)

30 DIM B\$(2,20)

40 NX=1:REM НОМЕР ПОСЛЕДНЕЙ

ЗАПОЛНЕННОЙ ЗАПИСИ

50 A\$(1,1)="ЛЕВ"

60 A\$(2,1)="?"

70 A\$(3,1)="?"

80 MX=1:REM НОМЕР ТЕКУЩЕЙ

ПРОСМОТРЕННОЙ ЗАПИСИ

90 IF A\$(2,MX)="?"GOTO 150

100 LPRINT A\$(2,MX); " ИЛИ ";

A\$(3,MX); " (1/2) "

110 INPUT LX

120 IF LX>2 THEN STOP

130 MX=B\$(LX,MX)

140 GOTO 90

150 LPRINT A\$(1,MX); " 1/0 "

160 INPUT LX

170 IF LX=1 GOTO 80



```

180 LPRINT "ВВЕДИТЕ НОВОЕ ИМЯ"
190 INPUT X$
200 LPRINT "ЧЕМ ОТЛИЧАЕТСЯ ОТ ";A$(1,M%)
210 INPUT Y$
220 LPRINT "A ";A$(1,M%)
230 INPUT Z$
240 A$(2,M%)=Y$
250 A$(3,M%)=Z$
260 B$(2,M%)=N%+1
270 B$(1,M%)=N%+2
280 A$(1,N%+1)=A$(1,M%)
290 A$(2,N%+1)="?"
300 A$(3,N%+1)="?"
310 N%=N%+2
320 A$(1,N%)=X$
330 A$(2,N%)="?"
340 A$(3,N%)="?"
350 GOTO 00
360 END

```

Пример работы программы

```

ЛЕВ 1/0
0

```

```

ВВЕДИТЕ НОВОЕ ИМЯ
ОКУНЬ
ЧЕМ ОТЛИЧАЕТСЯ ОТ ЛЕВ
ПЛАВАЕТ
А ЛЕВ
ХОДИТ
ПЛАВАЕТ ИЛИ ХОДИТ (1/2)
1
ОКУНЬ 1/0
0
ВВЕДИТЕ НОВОЕ ИМЯ
АКУЛА
ЧЕМ ОТЛИЧАЕТСЯ ОТ ОКУНЬ
ЕСТ ЛЮДЕЙ
А ОКУНЬ
НЕ ЕСТ
ПЛАВАЕТ ИЛИ ХОДИТ (1/2)
1
ЕСТ ЛЮДЕЙ ИЛИ НЕ ЕСТ (1/2)
1
АКУЛА 1/0
1
ПЛАВАЕТ ИЛИ ХОДИТ (1/2)

```

В. БОРИСОВ, С. ПОЛЯКОВ  
г. Саратов

## Использование языков высокого уровня на КУВТ-86

В статье описывается пакет программ, позволяющий без каких-либо изменений аппаратных средств значительно расширить возможности КУВТ-86 с Фокалом, обеспечивая:

работу учащихся на любом языке программирования (Паскаль, Си, Бейсик и т. д.) из входящих в состав базового программного обеспечения ДВК-2М;

подготовку текстов программ для БК-0010 на ДВК с помощью одного из текстовых редакторов, имеющихся на ДВК (EDIT, K52 и др.);

выполнение на БК-0010 программ, написанных на языках Си и Паскаль;

просмотр, распечатку и редактирование программ, созданных на БК-0010 с помощью его штатного интерпретатора языка Фокал, без их загрузки в БК-0010;

передачу в БК-0010 произвольного текста из ДВК (до 250 строк по 64 символа в строке) с возможностью дальнейшего его просмотра на экране БК или записи на магнитную ленту.

Программы пакета написаны на языке Си, для обмена информацией между БК-0010 и ДВК-2М используются несколько доработанные стандартные программы работы с классом (PUTBK.SAV, GETBK.SAV, PUTPBK.SAV). Целью доработки программ PUTBK.SAV и GETBK.SAV было обеспе-

чение возможности их использования в командных файлах, так как стандартно поставляемые варианты этих программ допускают ввод номера микро-ЭВМ и имени файла лишь с клавиатуры дисплея. В пакет входят следующие программные средства:

CONVER.SAV — программа, позволяющая преобразовать принятый из БК и записанный в файл на диске текст программы из формата абсолютной загрузки, используемого при передаче по последовательному каналу, в обыкновенный текст, который можно вывести на экран дисплея или печатающее устройство, редактировать с помощью какого-либо текстового редактора на ДВК. При использовании соответствующего ключа эта программа может убрать в начале каждой строки ее номер, символ С и следующий за ним пробел.

RECONV.SAV — программа, позволяющая преобразовать текст программы на Фокале, созданный с помощью редактора на ДВК, в формат абсолютной загрузки, используемый при передаче файлов из ДВК в БК-0010, т. е. выполняет работу, обратную производимой CONVER.SAV.

EDFOC.SAV — программа, которая воспринимает произвольный текст и делает из него «программу» на Фокале, т. е. в начале каждой строки текста вставляет номер строки в соответствии с правилами

Фокала, затем ставит символ С (признак комментария в Фокале) и пробел.

Для обеспечения выполнения в микро-ЭВМ БК-0010 программ, написанных на Паскале, на основе программы, приведенной в [1], был разработан эмулятор системных запросов ввода-вывода. Для работы в БК-0010 программ на языке Си была модифицирована библиотека подпрограмм ввода-вывода одной из версий системы программирования на языке Си для операционной системы RT-11. В эту библиотеку были также добавлены функции, позволяющие использовать графические возможности БК-0010.

К достоинствам описываемого пакета следует отнести предоставление разработчикам обучающих программ для КУВТ-86 возможности использования языка Си, позволяющего писать компактные и быстродействующие программы. Причем разработка программ и их трансляция могут проводиться и на более мощных, чем ДВК-2М, машинах типа СМ-1420, «Электроника 100-25», «Электроника-79».

Использование пакета при работе с отличными от Фокала алгоритмическими языками предполагает следующие этапы работы.

1. На БК-0010 ученик набирает текст программы на любом языке программирования (Си, Паскаль, Бейсик, Фортран и др.), причем в соответствии с синтаксисом Фокала в начале каждой строки ставится фиктивный номер оператора и символ С (т. е. с точки зрения Фокала строка представляет собой оператор COMMENT).

2. Этот текст записывается на дискету ДВК с помощью GETBK.SAV.

3. CONVER преобразует полученный файл из внутреннего формата интерпретатора Фокала в нормальный текст, убирая номера строк, символ С и следующий за ним пробел. Программа приобретает вид, привычный для языка, на котором она составлялась.

4. Программа транслируется компилятором, работающим на ДВК.

5. Программы, написанные на языках Си и Паскаль и скомпонованные в формате абсолютной загрузки, передаются с помощью PUTBK.SAV в БК-0010 для выполнения. Если же программа написана на каком-либо другом языке, придется, к сожалению, позвать ученика к ЭВМ учителя и продемонстрировать ему, как его программа работает (или не работает) на ДВК.

6. Если в ходе трансляции или выполнения программы обнаружены ошибки, то программа может быть исправлена непосредственно на ДВК (если требуется помощь учителя) или ее текст может быть отправлен обратно в БК-0010 для самостоятельной работы ученика. Если программа была моди-

фицирована на ДВК и ее текст отличается от текста в файле формата обмена с БК-0010, а возникла необходимость переслать ее в БК, то с помощью EDFOC.SAV к ее строкам снова «пришиваются» фиктивные номера, символ С, она преобразуется с помощью RECONV.SAV, и полученный файл с помощью PUTBK.SAV пересылается в БК-0010.

Операции 2—5 выполняются без участия оператора под управлением командного файла.

Использование описанного пакета дает разнообразные преимущества. В числе основных — более широкие возможности создания обучающе-контролирующих программ по различным предметам и использования КУВТ-86 как основы компьютерных клубов для развития научно-технического творчества молодежи.

В частности, при работе с обучающе-контролирующими программами, написанными на Фокале, Бейсике или Т-языке системы «Рига», ученик имеет доступ к тексту контролирующей его программы, что в достаточной степени делает бессмысленной оценку результатов проверки знаний. Программы, написанные на Си или Паскале, полностью исключают возможность несанкционированного доступа к контролирующей программе и данным в ЭВМ.

Другой аспект — быстрое действие. Сравним, например, программу KARR, созданную в Латвийском государственном университете, и программу STARC, созданную учеником X класса, активным членом компьютерного клуба «Алгоритм» средней школы № 42 г. Саратова Романом Мозговым. KARR, написанная на Фокале, условно изображает вид созвездий на небе в определенное время, установленное пользователем. STARC, написанная на Си, обучает находить основные созвездия и контролирует качество обучения. При использовании KARR из-за низкого быстродействия интерпретатора Фокала ученик пассивен: он вынужден ждать, пока наконец машина нарисует все созвездия, а это продолжается более пяти минут.

В программе STARC изображение звездного неба появляется моментально. В режиме мультипликации происходит обучение нахождению созвездий на небе, которое имеет на экране привычный для глаза вид. За счет высокого быстродействия обучение можно повторять неоднократно. Качество обучения контролируется так: на экране появляется вид звездного неба, где изучаемые созвездия обозначены цифрами, причем всегда новыми, и ученик должен ответить, какой номер имеет каждое созвездие. При желании обучение или попытку определить

все созвездия можно повторить.

Высокое быстродействие, обеспеченное использованием при создании программы компилятора языка Си, позволяет сосредоточить внимание ученика на содержательных моментах обучения и его самостоятельных действиях, а не на ожидании реакции ЭВМ.

Необходимо учесть и тот факт, что учащиеся при использовании описываемого пакета знакомятся с одними из наиболее эффективных и все чаще используемыми в последнее время языками — Си и Паскалем. Пакет прошел апробацию в течение двух лет в школе № 42 г. Саратова и в настоящее

время распространяется Саратовским институтом усовершенствования учителей в школах города и области. Опыт работы показал, что ученики достаточно успешно осваивают программирование на Си и, освоив этот язык, уже не хотят возвращаться к Фокалу.

## Литература

1. Завилов В. Н., Константинов М. Ю., Померанец М. В. Программирование на языке Паскаль для микро-ЭВМ «Электроника БК-0010» // Микропроцессорные средства и системы. 1987. № 1.

Е. КОЦЮБА

## Сделайте АОС сами!

Сегодня большему числу читателей «ИНФО» требуются стандартные ППС. Однако очень часто необходимых ППС еще нет либо они недоступны в силу различных причин. Если же ППС и имеются, то они могут не удовлетворять читателя своей негибкостью и «закрытой» структурой, чрезмерной сложностью или громоздкостью. На разработку же собственных программ уходит много сил и времени. А как хотелось бы уже сегодня, не дожидаясь помощи «сверху», на деле начать осуществление пресловутого принципа индивидуального подхода (для класса в 40 человек за 45 мин!). Использование ПЭВМ даже для достаточно примитивного, но регулярного опроса приводит к значительному повышению качества знаний и сокращению рутинной, нетворческой части работы учителя. Однако применение традиционных языков программирования для подготовки таких ППС ежедневного использования малоэффективно.

Ниже будет изложена идеология построения простой системы автоматизированного обучения и контроля. Основными ее достоинствами являются простота, полная открытость для пользователя и разработчика, возможность реализации на любой ПЭВМ, возможность работы в минимальной конфигурации с последующим значительным расширением, полная свобода для творчества.

Поскольку традиционный подход к контролю знаний и обучению укладывается в простую схему «вопрос — ответ» или «порция информации — вопрос — ответ», то набор функций или команд реализующей его системы невелик, а структура примерно такова: сценарий опроса или урока, состоящий из последовательного набора

команд, и интерпретатор сценария, написанный на любом из языков программирования, например Бейсике.

Сценарий напоминает командный файл для ДВК и создается обычным редактором текста. В сценарий входят тексты вопросов и команды, задающие образцы правильных ответов и варианты их анализа, вызывающие новый сценарий и т. д. В каждой строке может находиться только одна команда. Признаком команды является наличие, например, символа «#» в первой позиции. Строка без признака команды — это строка текста, подлежащая выводу на дисплей.

Более детальное структурирование удобно провести одновременно с описанием функционирования системы.

Блок инициализации и загрузки сценария считывает файл сценария в текстовый массив, где каждой строке файла соответствует элемент массива. При инициализации обнуляются счетчики вопросов, правильных ответов, количества баллов. Номер текущей строки устанавливается на единицу, и управление передается анализатору строки.

Анализатор проверяет текущую строку на наличие признака команды; при отсутствии одного строка выводится на дисплей и увеличивается на единицу номер текущей строки, иначе определяется имя команды и происходит переход на подпрограмму обработки этой команды.

Подпрограммы обработки команд считывают операнды, следующие за именем команды, проверяют их допустимость, выполняют соответствующие действия и затем увеличивают номер текущей строки (на единицу или другое число в зависимости от



команды; например, для команды, дающей задание типа «исправь строку», требуются две строки текста).

Ввод ответа осуществляется одним из соответствующих блоков программы. В случае неправильного ответа блок эмоциональной оценки выдает сообщение типа «неверно» или «опять неправильно» и повторяется ввод ответа, а при верном ответе может быть сообщение «верно», «молодец!» и т. п. В случае неправильных ответов имеет смысл повторять вопрос не больше трех раз, а затем дать обучаемому верный ответ, если только речь не идет о строгой контрольной работе.

Оценка занятия производится в соответствии с долей заработанных за ответы баллов. Например, за правильный ответ с первой попытки начисляется пять баллов, на ответ со второй попытки — три, с третьей — два. Разную сложность вопросов можно учесть введением в команды ввода ответа соответствующего операнда, характеризующего «вес» вопроса. При наличии

в ПЭВМ часов возможен также учет времени, затраченного на поиск ответа, с соответствующей эмоциональной и балловой оценкой.

Наиболее продуктивным видом вопроса является вопрос типа «исправьте строку». При этом ученику дается возможность отредактировать строку с использованием клавиш перемещения курсора. При сравнении отредактированной строки с образцом ответа возможно возникновение проблем с числом пробелов, поэтому необходимо использовать дополнительную подпрограмму, удаляющую пробелы из строки.

В конце занятия определяется итоговая оценка и может даваться возможность один или несколько раз повторить занятие, а также возможна запись результатов в банк данных.

Вот простой пример сценария.

#Z 9 КЛАСС, УРОК 1, ВАРИАНТ 2  
КАКОЙ ПРЕДМЕТ МЫ НАЧИНАЕМ  
ИЗУЧАТЬ?

1 — ФИЗИКУ

2 — ПРОГРАММИРОВАНИЕ

3 — ОСНОВЫ ИНФОРМАТИКИ И ВЫЧ.  
ТЕХНИКИ

#A3

КАКОЙ ЯЗЫК ПРОГРАММИРОВАНИЯ  
МЫ БУДЕМ ИЗУЧАТЬ

В ПРАКТИКУМЕ ПО ОСНОВАМ ИН-  
ФОРМАТИКИ?

#АБЕЙСИК

НАПИШИТЕ ПРАВИЛЬНО СЛЕДУЮ-  
ЩИЕ ТЕРМИНЫ:

#I КАМПЬЮКТЕР

КОМПЬЮТЕР

#I АЛГАРИТМ

АЛГОРИТМ

#E

Команда Z выводит на дисплей приветствие машины, параметры задания и переводит АОС в режим ожидания ответа. Команда A в первом случае получает ответ в виде одного символа (требуется нажатие только одной клавиши), во втором ввод ответа аналогичен вводу по оператору Бейсика INPUT (возможно редактирование текста ответа, в конце необходимо нажать клавишу ENTER или BK). Команда I в первой строке имеет предлагаемый вариант текста, во второй строке — правильный ответ.

Вот примерный список имен необходимых команд, которые не претендуют на мнемоничность и выбраны односимвольными из соображений простоты работы анализатора текущей строки.

Z — начало занятия, вывод параметров

E — конец занятия, вывод оценки

A — ввод ответа — литерной величины

I — ввод ответа типа «исправленная строка»

P — пауза

D — задание макрокоманды

M — вызов макрокоманды

R — возврат из макрокоманды

C — вызов подпрограммы

X — рестарт сценария

L — загрузка нового сценария

G — ветвление

Из всех команд только первые три являются обязательными, остальные могут добавляться по мере необходимости.

От редакции. Вышеописанная АОС относится к числу самых простых, однако работа по ее созданию может оказаться весьма полезной — в первую очередь для самого создателя. Подобные проекты можно порекомендовать для выполнения слушателям факультативов по программированию.

О. БОКСЕР, докт. мед. наук, А. ВАСИЛЬЧЕНКО, А. ПРИЯТКИН

## Реакциометрические обследования на учебных ПЭВМ

В условиях активного применения информационных технологий значение непрерывного контроля психофизиологических характеристик человека, ведения его физиологического паспорта резко возрастает. Это вызвано потребностями общества в качественном профотборе и профориентации, медицинской и социальной коррекции [1].

В психофизиологическом тестировании нуждаются самые широкие слои населения всех возрастных групп. Психофизиологический контроль необходим работникам множества профессий: водителям всех видов транспорта, операторам электростанций, диспетчерам железнодорожных узлов и любых других управляемых объектов подобного типа, спортсменам, пожарным и др. Без хорошо поставленной службы непрерывного психофизиологического контроля невозможно дальнейшее развитие общества. Психофизиологические данные каждого человека должны учитываться при выполнении любой полезной деятельности в обществе.

Реакциометрия человека относится к фундаментальным направлениям исследования психофизиологических характеристик человека. Представляемая в элементарном варианте как измерение ответной реакции человека на предъявляемый ему раздражитель, она позволяет тестировать умственное и физическое утомление человека, воздействие на него стрессовых ситуаций, определять тип его центральной нервной системы (тип высшей нервной деятельности) [2]. Реакциометрия человека лежит в основе многих практических методов профотбора, например, космонавтов и летчиков, моряков и операторов АЭС и др.

Оснащение учебных заведений страны персональными ЭВМ, и локальными сетями на их основе, предоставляет возможность про-

ведения массовых реакциометрических обследований. Вычислительных, цветоцветовых и звуковых ресурсов учебных ПЭВМ массового применения вполне достаточно для выполнения реакциометрического тестирования с достаточной точностью и построения широкой гаммы тестов различного физиологического профиля. На современных учебных ПЭВМ возможна реализация реакциометрических тестов, ориентированных на пользователя-непрофессионала, которому в данном случае становится доступным проведение обследований, ранее выполняемых только в специально оборудованных лабораториях специально подготовленным персоналом. При компьютерном тестировании исключаются многие трудоемкие операции, такие, как, например, статистическая обработка результатов испытаний.

Ведение психофизиологического паспорта человека вполне может быть «доверено» учебным ПЭВМ.

В последние десятилетия сложилось сложное положение с психофизиологическими приборами и их использованием, практически не проводилась массовая психофизиологическая аттестация, имеющая фундаментальное значение для профориентации и профотбора. Эта ситуация может и должна быть изменена при эффективном применении имеющегося парка учебных и бытовых ПЭВМ. Открывающаяся на этой методической основе возможность массовых психофизиологических обследований в масштабах страны имеет огромное социальное и научное значение.

Проведение таких обследований важно начинать уже с детских дошкольных учреждений, возможно, в форме компьютерных игр, что, естественно, нисколько не повредит методике обследования и положительно скажется на психоэмоциональном состоянии

ребенка. Обследования должны последовательно осуществляться в школах, средних и высших учебных заведениях, на производстве. Естественно, что для выполнения таких задач и приобретения необходимой квалификации, в первую очередь педагогами и медиками, должен быть соответствующим образом откорректирован учебный процесс в университетах, педагогических и медицинских вузах.

В Шуйском государственном педагогическом институте на базе межкафедральной психофизиологической лаборатории (МПФЛ) ведется большая работа в данном направлении, к участию в которой привлекаются студенты. Институт располагает для этих целей необходимым арсеналом ПЭВМ.

Научно-организационной основой разработки и внедрения программного обеспечения реакциометрических обследований на учебных ПЭВМ является комплексное взаимодействие на базе МПФЛ несколько кафедр и учебных курсов института: кафедр математики, психологии, возрастной физиологии, спортивной психологии, психофизиологии труда (в рамках факультетов физического воспитания и индустриально-педагогического), отдела технических средств обучения.

Разработанные программы фундаментальных хронореакциометрических обследований внедрены в учебный процесс и научные исследования. Внедрение в учебный процесс разработанного программного обеспечения осуществляется на основе взаимосвязанного учебного плана.

Первый этап заключается в овладении студентами младших курсов института основами информатики, программирования, управления учебными ПЭВМ и локальными сетями на их основе. В процессе приобретения базовых знаний практикуется фрагментарное использование разработанного программного обеспечения хронореакциометрии.

Реакциометрическое тестирование проводится в спортивных секциях института и спортшколах города, а также среди учащихся общеобразовательных школ, проходящих учебно-вычислительную практику в институте.

На втором этапе проводятся конкретные лабораторные исследования в соответствующих курсах спортивной и возрастной физиологии, психофизиологии труда и т. д.

На третьем этапе студенты, особенно члены НСО, привлекаются к разработке курсовых и дипломных проектов на базе МПФЛ в союзе с другими кафедрами. Методической основой этих работ является применение компьютерной техники в психофизиологических обследованиях.

#### Л и т е р а т у р а

1. Громов Г. Р. Гуманитарные основы информационной технологии. М., 1988. С. 36.
2. Боксер О. Я. Хронометраж физиологический // Большая медицинская энциклопедия. М.: Медицина. 1986. 3-е изд. Т. 27. С. 401—404.

## Файловый монитор FILMON для ПЭВМ «Микроша»

По аналогии с дисковой операционной системой FILMON можно назвать кассетной операционной системой. Он позволяет реализовать все традиционные функции оперирования файлами с последовательным доступом. Аппаратной базой FILMON служит кроме самой ПЭВМ модифицированный кассетный магнитофон.

Суть модификации магнитофона в следующем: во-первых, он оборудуется датчиком, регистрирующим каждый оборот подкассетника, что позволяет компьютеру определять, какое примерно место магнитной ленты находится напротив магнитной головки. Во-вторых, добавляется (если ее не было) электро-механическая система, позволяющая компьютеру переключать режимы магнитофона, подавая электрические сигналы.

Для пользователя работа на компьютере, оснащенном файловым монитором, практически не отличается от работы на компьютере с дисководом. Разница только во времени ожидания исполнения приказов.

Монитор, созданный В. Кузнецовым, Л. Лаптевым, А. Лапушкиным, А. Шведовым, В. Дворцевым, позволяет:

- работать с каталогами двух кассет одновременно;
- просматривать каталоги кассет;
- искать, читать и записывать на кассету файлы как в диалоговом режиме, так и под управлением программ пользователя;
- удалять, переименовывать, копировать файлы;
- запускать программы указанием их имен (FILMON сам ищет файл на ленте, считы-

вает его и запускает на исполнение);

создавать на кассете МК-60 крупные базы данных (до 1,8М байт) с оперативным поиском информации (среднее время доступа — 1—2 мин);

после открытия кассеты автоматически запускать программу, имя которой указано в поле AUTOEXEC каталога.

FILMON предоставляет возможность работы с файлами на ленте всем программам, использующим ввод-вывод на магнитофон. Доработка для этой цели существующих программ (в том числе интерпретаторов Бейсика, всевозможных редакторов, ассемблеров и т. д.) минимальна.

FILMON написан на ассемблере микропроцессора INTEL8080/КР580ВМ80 и занимает в памяти около 4К байт (эта величина зависит от версии). Кроме того, для хранения каталогов кассет он использует 1К байт и еще 1К байт для своих рабочих областей. Сам монитор может храниться и в ПЗУ.

Для ПЭВМ «Микроша» FILMON реализован в отдельном блоке, подключаемом к разъему «внутренний интерфейс» компьютера и к магнитофону.

FILMON написан с учетом требований структурного программирования и состоит из двух основных блоков: ядра системы и командного процессора. Последний является «надстройкой» над ядром, в котором в виде подпрограмм и функций реализованы все функции системы FILMON, а также все подпрограммы, необходимые для построения си-

стем, работающих с файлами на ленте. Вместо командного процессора можно использовать программу SCREEN-COMMANDER, позволяющую работать с файловой системой в режиме диалога через меню. Таким образом, система является легко расширяемой и модифицируемой. FILMON может быть установлен на любой компьютер, изготовленный на базе микропроцессора КР580ВМ80, использующий стандартный системный монитор и имеющий в своем составе параллельный порт ВВ55 и интервальный таймер ВИ53.

Основные характеристики системы FILMON:

максимальное число файлов на одной стороне кассеты — 32;

максимальный размер файла — 32К байт; возможность обслуживания одновременно двух кассет (на двух магнитофонах или на одном с двумя лентопротяжками).

Кроме того, подключаемый блок с системой добавляет «Микроше» 16К байт ОЗУ.

С использованием монитора разработана локальная сеть, способная объединить до 30 «Микрош». При этом они могут выступать как один накопитель информации емкостью до 60М байт со средним временем доступа 1—2 мин, что позволяет создавать на базе такого комплекса весьма серьезные базы данных.

В следующих номерах журнала будет рассказано о необходимых для использования FILMON доработках магнитофона и ПЭВМ.

Эта статья предназначена новым читателям нашего журнала — надеемся, они у нас появились.

Д. КОРОБИЦЫН

## Изучаем Бейсик за полчаса

### Введение

Здравствуйте, уважаемый читатель!

Название статьи не шутка. В течение 30 мин действительно можно постигнуть азы программирования. Надеюсь, что вы уже знакомы с клавиатурой и знаете, что такое курсор.

К общению с ЭВМ приступайте только после прочтения шага 4.

Итак, время пошло. Начинаем...

### ШАГ 1

Вы хотите научиться программировать, иначе говоря, составлять программы. Что же это такое?

Программа — несколько команд, которые должна выполнить ЭВМ. Эти команды принято называть операторами. В языке Бейсик операторы нумеруют, чтобы ЭВМ могли понять, в какой последовательности их выполнять. Нумерация обычно идет с шагом 5 (5, 10, 15, 20,...) или 10 (10, 20, 30,...), чтобы в случае необходимости между любыми двумя операторами можно было вставить еще несколько.

### ШАГ 2

ЭВМ — электронно-вычислительная машина, значит, она предназначена для вычислений. Давайте вычислим квадратный корень из пяти. Так как на клавиатуре нет привыч-

ного обозначения знака квадратного корня, то используем его условное обозначение — `SQR`. После него в скобках записывается число, из которого нужно извлечь корень: `SQR (Y)`.

Программа 1 состоит из одного оператора. Выполнив ее, ЭВМ присвоит переменной `X` значение квадратного корня из 5.

```
Программа 1
10 X=SQR(5)
```

Вычислить можно не только квадратный корень, но и любое другое выражение. Записывают его как обычно, только вместо знака умножения используется `*` (звездочка), вместо деления — `/` (наклонная черта), вместо возведения в степень — `**` (две звездочки), вместо десятичной запятой — точка.

Программа 2 вычисляется и присваивает переменной `X` значение выражения, стоящего после знака равенства.

62 Программа 2  
10 X=2.5\*3.18+(125-21/6.8)\*\*3

Порядок выполнения действия обычный.

### ШАГ 3

Выполняя программу 1, ЭВМ вычислит значение корня из 5 и присвоит его переменной `X`, но на экране ничего не появится. А нам бы хотелось узнать, чему же все-таки равен этот корень.

Чтобы выводить информацию на экран, используется оператор `PRINT` — печать.

```
Программа 3
10 X=SQR(5)
20 PRINT X
```

После выполнения этой программы на экране появится число 2.26784 — это и есть значение корня квадратного из 5 с точностью до 5 знаков\*.

Оператор `PRINT X` выводит значение переменной `X` на экран.

Если запустить следующую программу:

```
Программа 4
10 X=SQR(5)
20 PRINT "X=";X
```

то на экране появится  
`X=2.26784`.

То, что набрано в кавычках, оператор `PRINT` печатает без изменений. В нашем случае это было `X=`, но вообще может быть любой текст. Знак `;` (точка с запятой) указывает ЭВМ, что значение переменной `X` надо печатать сразу же после текста в кавычках, т. е., начав выполнять оператор `PRINT`, ЭВМ напечатает текст в кавычках (на экране

появится `X=`), а затем значение переменной `X`, не пропуская ни одной позиции.

### ШАГ 4

А теперь — за клавиатуру. Наберем на ней `NEW` и нажмем клавишу «ввод». Команда `NEW` (новый) указывает, что нужно очистить память от всех программ, введенных раньше, и приготовиться к вводу новой программы.

Наберите программу 4. После набора каждого оператора нужно нажимать клавишу «ввод».

Действуйте!

Набрав в программу, наберите `LIST` и снова «ввод». На экран выведется текст набранной вами программы. Команда `LIST` (список) указывает ЭВМ, что нужно вывести текст программы из памяти на экран.

Сравните текст на экране с текстом программы 4. Если вы обнаружите ошибку в одном из операторов, наберите его еще раз, только правильно. После этого можете повторно просмотреть текст командой `LIST`.

Вообще, для исправления или замены одного оператора в уже набранной программе надо набрать исправленный или новый оператор с таким же номером.

Для запуска программы на выполнение наберите `RUN` (и, как обычно, нажмем клавишу «ввод»). Эта команда указывает ЭВМ, что надо приступить к исполнению программы, находящейся в памяти.

Внимание! После набора каждого оператора или команды не забывайте нажимать клавишу «ввод».

### ШАГ 5

Поздравляю вас с первой набранной и запущенной программой.

Теперь давайте попробуем сделать так, чтобы наша программа могла вычислять квадратный корень не только из пяти, но и из любого другого (неотрицательного) числа. Это можно сделать, например, написав

```
10 X=SQR (Y)
```

(`Y` — переменная). Значение квадратного корня из `Y` будет присвоено в этом операторе переменной `X`, значение которой мы выведем на экран оператором `PRINT`.

Значение `Y` необходимо задать. Это можно сделать так:

```
Программа 5
5 Y=12
10 X=SQR (Y)
20 PRINT "X=";X
```

Однако при таком задании значения переменной `Y` мы мало что выигрываем по сравнению со следующей программой:

```
Программа 6
10 X=SQR (12)
20 PRINT "X=";X
```

\* Увы:  $2.26784^2=5.1431$ , а  $\sqrt{5}=2.23\dots$ , но это вина не автора статьи, а проектировщиков его компьютера. — *Примеч. ред.*



И в программе 5, и в программе 6 каждый раз, когда нужно вычислить корень из нового числа, надо перенабирать один из операторов. Чтобы этого избежать, используется оператор INPUT — ввод. Он указывает ЭВМ, что нужно запросить значение, и оно будет введено с клавиатуры.

```
Программа 7
5 INPUT Y
10 X=SQR(Y)
20 PRINT "X="; X
```

При запуске этой программы (команда RUN) на экране появится вопросительный знак. Это ЭВМ предлагает вам ввести число Y с клавиатуры.

Наберите число, нажмите клавишу «ввод». На экране появится значение квадратного корня из этого числа.

Чтобы вычислить корень из другого числа, надо опять дать команду RUN и после того, как на экране появится вопросительный знак, ввести число.

Не забывайте: дать команду — значит набрать ее и нажать «ввод»; ввести число — значит набрать его и нажать «ввод».

Чтобы занести в память ЭВМ программу 7, не нужно набирать ее полностью. Можно изменить уже имеющуюся в памяти программу 6. Для этого надо набрать новый оператор 5 и перенабрать оператор 10, причем можно сначала исправить оператор 10, а затем ввести оператор 5. Благодаря наличию нумерации операторы разместятся в памяти ЭВМ в том порядке, в каком нужно.

Вы можете это проверить, просмотрев текст программы командой LIST. Для запуска программы на выполнение пользуйтесь командой RUN.

Действуйте!

## ШАГ 6

Мы написали программу, которая вычисляет корень из любого неотрицательного числа. Ну, а что же произойдет, если мы все же введем отрицательное число?

Попробуйте.

ЭВМ выдает сообщение об ошибке. Постараемся этого избежать: будем проверять, отрицательное число введено или нет.

Для этого используется оператор IF... THEN... (если... тогда...).

Оператор 7 IF Y<0 THEN 30 будет выполнять следующее: если Y<0, то перейти к выполнению оператора с номером 30; если нет, то перейти к выполнению оператора, следующего за IF...THEN... Оператора номер 30 у нас нет, добавим его:

```
Программа 8
5 INPUT Y
7 IF Y<0 THEN 30
10 X=SQR(Y)
20 PRINT "X="; X
30 PRINT "ПРИВЕТ!"
```

Оператором 30 мы сделали оператор PRINT. Текст в кавычках будет выведен на экран.

Чтобы перейти от программы 7 к программе 8, нужно набрать новые операторы 7 и 30.

Программа 8 работает так:

если мы введем отрицательное число, то «сработает» оператор 7, произойдет переход на выполнение оператора 30, который напечатает на экране «Привет!»;

если мы введем неотрицательное число, то перехода в операторе 7 не произойдет, будет вычислен и напечатан квадратный корень из введенного числа, а после результата будет напечатано «Привет!».

Действуйте!

## ШАГ 7

Давайте избавимся еще от одного недостатка нашей программы.

Каждый раз, чтобы вычислить корень еще из одного числа, необходимо давать команду RUN. 63

Используем оператор GOTO (перейти на ...).

Оператор 25 GOTO 5 указывает, что после него надо выполнять не следующий за ним оператор (номер 30), а оператор, номер которого указан после GOTO, в нашем случае оператор номер 5.

```
Программа 9
5 INPUT Y
7 IF Y<0 THEN 30
10 X=SQR(Y)
20 PRINT "X="; X
25 GOTO 5
30 PRINT "ПРИВЕТ!"
```

Эта программа будет запрашивать у вас число, выводя вопросительный знак, вычислять (оператор 10) и печатать (оператор 20) квадратный корень из него, затем будет переходить (оператор 25) на ввод еще одного числа.

Так будет продолжаться до тех пор, пока вы не введете отрицательное число. Тогда сработает оператор 7. Он передаст управление на оператор 30, который напечатает «Привет!», и программа закончит работу.

Проверьте!

## ШАГ 8

Теперь давайте выведем на экран таблицу квадратных корней чисел от 1 до 10.

Если мы воспользуемся программой 9, то нам придется десять раз вводить число с клавиатуры. Десять чисел ввести еще можно, ну а если нам понадобится таблица квадратных корней чисел от 1 до 100 или от 1 до 1000?

Чтобы не вводить так много чисел, можно воспользоваться операторами цикла FOR и NEXT. Они показывают, что все операторы,

стоящие между ними, должны быть выполнены несколько (сколько именно — задает оператор FOR) раз.

```
Программа 10
5 FOR Y=1 TO 10 STEP 1
10 X=SQR(Y)
20 PRINT "SQR( ";Y;" )="; X
25 NEXT
30 PRINT "ПРИВЕТ!"
```

Оператор 5 FOR Y=1 TO 10 STEP 1 (от Y=1 до 10 с шагом) показывает ЭВМ, что переменной Y надо сперва присвоить значение 1. Затем выполнить все операторы, находящиеся между FOR и NEXT. В нашем случае это операторы 10 и 20, при этом будет вычислена и напечатана одна строка таблицы. Затем ЭВМ должна увеличить значение Y на 1 (именно так расшифровывается выражение «STEP 1») и опять выполнить операторы 10 и 20, т. е. напечатать еще одну строку таблицы. И так повторять до тех пор, пока значение Y не станет больше 10, т. е. пока Y не пробежит все значения от 1 до 10 с шагом 1. На экране в результате этого будет напечатана интересующая нас таблица.

Чтобы перейти от программы 9 к программе 10, надо перенабрать операторы 5 и 25 и уничтожить оператор 7. Для уничтожения надо набрать 7 и нажать «ввод». Для проверки набранного текста пользуйтесь командой LIST.

Если мы захотим получить другую таблицу, то надо перенабрать оператор 5, задав нужные параметры. Например, если нужна таблица от 0 до 1 с шагом 0.1, то оператор 5 будет выглядеть так:

```
5 FOR Y=0 TO 1 STEP 0.1
```

Слово STEP в операторе FOR можно и не указывать. В этом случае шаг (как говорится, по умолчанию) будет равным 1.

### Заключение

Ну, вот и всё. Посмотрите на свои часы, должно было пройти около получаса. На-

деюсь, что вы не пожалеете о времени, которое вы затратили сверх обещанных 30 минут.

Если программирование сравнить с игрой в шахматы, то вы теперь знаете главные правила игры. Если есть желание, то со временем вы постигнете и более сложные аспекты.

Для дальнейшего изучения языка Бейсик обращайтесь к книгам Ю. Л. Кеткова «Программирование на Бейсике» и «Диалог на языке «Бейсик» для мини- и микро-ЭВМ».

Счастливого пути к знанию!

### Памятка

```
PRINT-----вывод на экран
10 PRINT "X=";X
напечатать: X=значение X'
30 PRINT "ПРИВЕТ!"
напечатать: ПРИВЕТ!

INPUT-----ввод с клавиатуры
15 INPUT Y
ввести значение Y

IF...THEN-----проверка условия
если...тогда
10 IF Y<0 THEN 30
15 ...
если Y<0, то выполнить
оператор номер 30. В
противном случае выполнить
оператор номер 15

GOTO-----безусловный переход
25 GOTO 5
перейти к выполнению
оператора номер 5

FOR...TO-----оператор цикла
от...до
5 FOR Y=1 TO 10 STEP 1
...
25 NEXT

NEXT-----конец цикла
следующий
меняя Y от 1 до 10 с
шагом 1, выполнять для
каждого значения Y все
операторы, стоящие между
операторами номер 5 и 25
```

64

## Новые пособия

В издательстве «Знание» в помощь изучающим основы информатики и вычислительной техники вышло очередное экранное пособие — комплект цветных диапозитивов с сопроводительной брошюрой «Тенденции развития мирового парка ЭВМ». В пособии, включающем 18 диапозитивов, раскрыты тенденции развития мирового парка ЭВМ и рынка информационных услуг, освещены вопросы перспективного развития программного обеспечения и периферийных устройств ЭВМ, разрабатываемых крупнейшими фирмами западных стран и стран членов СЭВ.

Среди названий диапозитивов — «Структура мирового парка ЭВМ», «Масштабы производства персональных ЭВМ в США и Японии», «Эволюция применения бытовых компьютеров», «Динамика рынка программных средств» и др. Высокое качество диапозитивов, подробный сопроводительный текст к каждому из них, а также удобный формат всего комплекта выгодно отличают это экранное пособие от других. Пособие в основном предназначено для лекторов Всесоюзного общества «Знание» и преподавателей народных университетов, однако оно с

успехом может использоваться в процессе преподавания курса информатики в школах, техникумах, вузах. В ближайшее время в этой серии выйдут и другие комплекты — «Языки программирования для персональных компьютеров», «Микрокомпьютеры: устройство, программирование, использование», «Автоматизированные обучающие системы» и др.

Студия «Диафильм» выпустила диафильмы «Роль ЭВМ в современном обществе» и «Компьютер — детям».

Первый из них предназначен для кабинетов вычислительной техники и может быть использован в процессе преподавания курса информатики в IX классе общеобразовательной школы. Диафильм в увлекательной и красочной форме рассказывает о применении ЭВМ в различных областях науки, техники, народного хозяйства. Первая часть диафильма иллюстрирует применение ЭВМ в различных автоматизированных системах управления (АСУ). Отдельные кадры показывают примеры: автоматизированный станочный комплекс «Талка-500», управляемый ЭВМ и действующий на Ивановском заводе тяжелого станкостроения; автоматическое управление полетом межпланетных станций «Венера-15» и «Венера-16»; АСУ в сферах бытового обслуживания и сельского хозяйства и др.

Следующие части рассказывают о работе

ЭВМ в автоматизированных системах научных исследований (АСНИ) и системах проектирования (САПР), о решении самых разнообразных задач социально-экономического развития нашего общества.

Диафильмы можно рекомендовать для использования не только на уроках информатики и вычислительной техники, но и при проведении различных внеклассных занятий, связанных, в частности, с пропагандой основных направлений научно-технического прогресса.

Диафильм «Компьютер — детям» повествует об опыте работы детского компьютерного клуба Дома комсомольца и школьника Октябрьского района Москвы. Кадры диафильма знакомят с компьютерной техникой, которой оснащен клуб, ее возможностями для развития детского творчества. В доступной для детей форме диафильм иллюстрирует огромные возможности компьютерной техники в различных сферах человеческой деятельности и вместе с тем подводит к мысли о необходимости освоения азбуки этого сложного и увлекательного мира информатики.

Диафильм можно рекомендовать для показа детям младшего и среднего возраста при первоначальном ознакомлении с этой новой областью человеческих знаний.

Е. СМЫСЛОВ

65

## КЛУБ БК

# Отсчет времени на БК-0010.01

Известно, что БК-0010.01 не имеет таймера, и мало кто знает, что при определенных условиях один из системных регистров с шестнадцатиричным адресом FFC8 может выступать в роли датчика времени. Для этого необходимо:

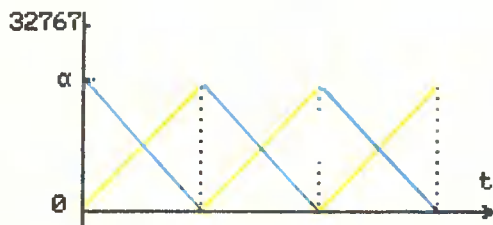
1) с помощью оператора `POKE &HFFC6, &Hn` задать в дополнительном коде начальное значение ( $n$ ) содержимого регистра FFC8, который мы в дальнейшем будем называть счетчиком;

2) оператором `POKE &HFFCA, &H70` выполнить запуск счетчика.

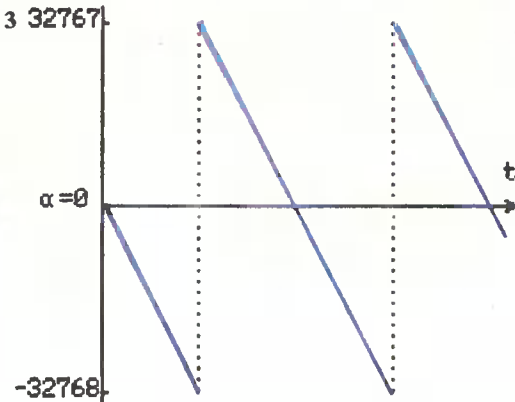
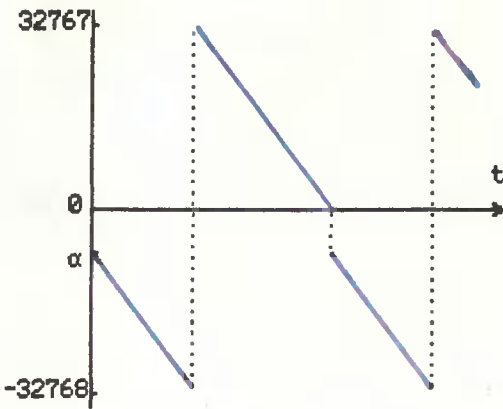
В результате содержимое счетчика станет изменяться во времени в соответствии с одним из графиков, изображенных на рис. 1—3. Считывание показаний счетчика производится операцией `PEEK (&HFFC8)`.

Нужно заметить, что после запуска счетчик работает непрерывно, не реагируя ни на какие

команды и операции Бейсика, даже на `STOP`, `END`, `NEW`. Только операция `POKE` с соответствующими адресами может изменить состояние счетчика. Следует иметь в виду также и то, что после включения БК его регистр FFCA, задающий начальное состояние счетчика, автоматически устанавливается на  $n=0$ . Поэтому если требуется задать именно это значение  $n$ , оператор `POKE &HFFCA, 0` можно опустить. Регистр FFCA тоже не вос-



1)



принимает команду NEW — в нем сохраняется прежнее значение  $a$ .

В случае, когда  $a=32767$  (&H7FFF), полный период работы счетчика по точным измерениям\* равен 89,657 с, откуда следует, что коэффициент пересчета показаний счетчика в секунды  $K=32767:89,657=365,47$ .

Теперь у нас есть все данные для построения таймеров, различных по исполнению и предназначению.

**Таймер со временем работы до 89 с и точностью отсчета в 1 с**

```

10 POKE &HFFCA,&H70
      'запуск счетчика
20 ? AT(0%,0%); "Время: ";
      ABS(INT(PEEK(&HFFCB)
      /365.47+.5)); "сек. "
30 GOTO 20
  
```

Операторы 20-й строки этой программы выдают на монитор время в секундах от нуля до 89, после чего счет идет в обратном направлении от 89 до нуля и т. д.

\* За 28 ч, т. е. за 100800 с.

**Таймер со временем работы до 179,3 с и точностью отсчета в 0,1 с**

Любой из этих таймеров можно использовать в обучающих программах для определения времени реакции на поставленный вопрос или времени, затраченного на получение правильного ответа, поскольку это время не превышает обычно нескольких десятков секунд.

```

10 POKE &HFFCA,&H70
20 T=PEEK(&HFFCB) ' чтение
      содержимого счетчика
30 IF T>0 THEN T=T-65536
40 ? AT(0%,0%); "Время: ";
      CSNG(INT(.05-T
      /36.547)/10); "сек. "
50 GOTO 20
  
```

С этой целью оператор, записанный выше в строке 10, располагают перед той частью обучающей программы, которая воспринимает и анализирует ответы, а остальные операторы, составляющие таймер, размещают в конце этой части.

**Таймер с произвольным временем работы и точностью отсчета в 0,01 %**

```

10 N%=0%
15 T%=0%
20 POKE &HFFC6,&H7FFF
      'альфа равно 32767
25 POKE &HFFCA,&H70
30 S%=T%
35 T%=PEEK(&HFFCB)
40 ? AT(0%,0%); "Время: ";
      CSNG(INT((32767*N%-T%)
      /3.6547+.005)/100);
      "сек. "
45 IF T%>S% THEN N%=N%+1%
50 GOTO 30
  
```

В отличие от предыдущих вариантов здесь подсчитывается и запоминается номер периода работы счетчика ( $N\%$ ), который затем используется для подсчета полного времени работы таймера. Такой алгоритм позволяет определять время с высокой точностью: погрешность не превышает 2 с за сутки. Поэтому этот таймер можно использовать, например, для построения программы электронных часов.

Если же высокая точность отсчета времени не нужна, можно воспользоваться более простой программой.

**Упрощенный таймер с произвольным временем работы**

Здесь цикл опроса счетчика замыкается

```

10 T=0
15 POKE &HFFCA,&H70
      'запуск счетчика
      в каждом цикле
20 ? AT(0%,0%); "Время: ";
      CSNG(INT(T/365.47+.5));
      "сек. "
25 T=T-PEEK(&HFFC8)
30 GOTO 15

```

на строку 15 программы; тем самым счетчик перезапускается заново в каждом цикле, а то, что он успевает за время цикла накопить, суммируется в строке 25. Это накопление по своей числовой величине от цикла к циклу почти не изменяется, в то время как значение переменной T постоянно возрастает, а значит, возрастает и разность порядков и время выполнения операции в строке 25 (за счет выравнивания порядков).

Рассмотрим несколько приложений. **Определение времени исполнения операторов и объема занимаемой ими памяти.**

Объем оперативной памяти БК, доступный пользователю при работе с Бейсиком, не столь велик, чтобы пренебрегать оптимизацией составляемой программы по числу занимаемых ею байтов. Нелишне знать и время выполнения каждого оператора в зависимости, например, от типа используемых данных.

Предлагаемая программа TRIAL, небольшая по объему и несложная по структуре, позволяет определять эти параметры.

```

8 POKE &HFFCA,&H70
9 FOR I%=0% TO 999%
10
...Участок для размещения
...испытываемых операторов
96
97 NEXT I%
98 ? "T=";INT(.5-
      PEEK(&HFFC8)/.36547)-
      140%; "мксек"; "L=";
      13728%-FRE(0%); "1=";
      200%-FRE(" ")
99 IF INKEY$=" " THEN
      EL 99

```

Между строками 10 и 96 этой программы размещают для исследования отдельные операторы Бейсика или небольшие их группы. Программа TRIAL выдает на монитор: время (T) исполнения операторов в микросекундах, число байтов (L), которые занимают эти операторы вместе с используемыми чис-

ловыми переменными, и отдельно — число байтов (I), занятых символьными переменными, если они имеются.

В процессе эксплуатации программы TRIAL выявились две особенности, проявляющие себя при определении времени T исполнения испытываемых операторов.

Первая заключается в «уходе» нуля на 8—10 мкс за час работы; вторая — в колебании на 2—3 мкс отсчетов T, полученных повторно при неоднократном нажатии пробела.

Для сравнительного анализа времени исполнения альтернативных операторов это обстоятельство не имеет значения. Но если требуются более точные оценки, рекомендуется следующее правило компенсации «ухода» нуля.

Ввести программу и пустить ее вхолостую, т. е. без заполнения промежутка 10—96 операторами. Нажимая несколько раз пробел, получить ряд отсчетов времени исполнения программы. Выбрать наименьший из них и сложить с константой 140 %, записанной в строке 98. Контрольным пуском программы убедиться в правильности сделанного исправления: наименьшее значение T должно равняться нулю.

Затем в промежуток 10—96 записать подвергаемые испытанию операторы, пустить программу и несколько раз повторить ее исполнение. Из ряда отсчетов T выбрать наименьший: он будет наиболее точным.

Если время исполнения исследуемых операторов в сумме превосходит 89657 мкс, то нужно либо разделить группу на отдельные операторы, либо внести некоторые изменения в программу, позволяющие увеличить в десять раз предел измерения времени. Вот эти изменения:

в строке 9 число 999 % заменить на 99 %; в строке 98 число .36547 заменить на .036547 и число 145 % — на 164 %.

Поскольку отсчет времени программой TRIAL производится с некоторым разбросом, не превышающим трех единиц младшего разряда, рекомендуется проводить повторные измерения, нажимая клавишу ПРОБЕЛ.

В приведенной ниже таблице даются некоторые результаты применения программы TRIAL.

Операторы	T	L	I
TX=12X+34X	93	36	0
T=12+34	1039	50	0
T!=12!+34!	1176	48	0
TX="12"+"34"	1395	50	4
? AT(0%,0%)	1630	28	0
? AT(0,0)	1942	42	0

67

```

FOR K=0 TO 8      11456  84  0
NEXT K

FOR K!=0! TO 8!  11814  78  0
NEXT K!

FOR K%=0% TO 8%  1444   60  0
NEXT K%

FOR K%=0% TO 8%  1743   62  0
NEXT

POKE 32767,      783   44  0
      32767

POKE &0077777,   68    34  0
      &0077777

POKE &H7FFF,     68    30  0
      &H7FFF

```

```

15 A%=INT(T*365.47-32766.5)
20 POKE &HFFC6,A% 'задание
                      альфа
25 BEEP 'звуковой сигнал
      (может задаваться и иной)
30 POKE &HFFCA,&H70
      'запуск счетчика с альфа
35 IF PEEK(&HFFC8)>0%
                      TH 25 EL 35

```

Промежуток времени T в этой программе может задаваться в пределах от 0,01 до 89,65 включительно с дискретностью не менее 0,01.

### Программа ЧАСЫ

В основе программы ЧАСЫ лежит еще один способ построения таймера на БК-0010.01. Он состоит в том, что после завершения каждого периода работы счетчика (от нуля до —32768) последний сбрасывается и запускается вновь с нуля операцией, записанной в строке 30. Возникающая при этом погрешность компенсируется увеличением константы, содержащейся в строках 20 и 35, до значения 89.6876. При этом ЧАСЫ дают за сутки погрешность, не превышающую 2 с.

```

10 POKE &HFFC6,0% 'альфа=0
15 INPUT "Введите время
      пуска часов: Ч,М,С";
                      Н%,М%,С%
20 T0=3600*Н%+60*М%+С%
                      -89.6876
25 CLS
30 POKE &HFFCA,&H70
      'пуск счетчика
35 T0=T0+89.6876
40 T%=PEEK(&HFFC8) 'съем
      показаний счетчика
50 Н%=T/3600
55 М%=T/60-Н%*60
60 С%=T-Н%*3600-М%*60
65 ? AT(17%,1%);Н% MOD 24%
      ;"ч.";М%;"мин.";С%;"сек."
70 GOTO 40

```

По запросу оператора INPUT, стоящего в строке 15, вводится начальное время пуска часов, при этом значения часов, минут и секунд отделяются друг от друга запятыми. В назначенное время часы запускаются нажатием клавиши ВВОД.

В. ЯКОВЛЕВ

68

Первая группа примеров (T=...) свидетельствует, во-первых, о том, что действия с целыми числами выполняются более чем на порядок быстрее, чем с вещественными, а во-вторых, что операции с числами ординарной длины (т. е. из 7 значащих цифр), помеченные символом «!», выполняются дольше, чем с числами удвоенной точности (с 17 значащими цифрами).

Вторая группа (? AT...) говорит о том, что не следует пренебрегать символом целого числа «%» в подобных операциях.

Третья группа (FOR... NEXT...) подтверждает уже известную нам истину о полезности использования, когда это возможно, целых чисел, но говорит еще и о том, что в операторе NEXT излишне указывать имя переменной, служащей параметром цикла.

Четвертая группа (POKE) показывает, что при использовании шестнадцатиричной адресации заметно экономится память БК. Кроме того, символ такой адресации &H трудно с чем-нибудь спутать, чего нельзя сказать о восьмеричной символике &O: букву O здесь легко принять за ноль, хотя последний и подчеркивается.

Даже этого небольшого числа примеров, по-видимому, достаточно, чтобы считать программу TRIAL весьма полезной.

Выдача сигнала через равные промежутки времени

```

10 INPUT "T=";T 'задание
      промежутка времени
      в секундах

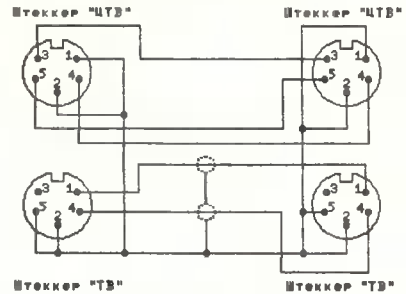
```

# Подключение БК к телевизорам ЗУСЦТ

Сейчас наиболее широко распространены цветные телевизоры типа ЗУСЦТ, и поэтому вопрос их доработки для работы с бытовыми компьютерами, а также доработки самих компьютеров волнует многих. Структурные и электрические схемы телевизоров ЗУСЦТ подробно рассмотрены в большом цикле статей журнала «Радио». Поскольку инструкцией этих телевизоров предусмотрена возможность их функционального наращивания, подключение к ним компьютеров БК не представляет каких-либо сложностей и вполне может быть выполнено в домашних условиях.

В зависимости от набора сервисных возможностей, требуемых при работе компьютера с телевизором, возможны различные варианты подключения БК. Оптимальным представляется следующий набор реализуемых функций: возможность работы компьютера как с черно-белым, так и с цветным изображением, вывод сигнала звукового сопровождения БК на усилитель НЧ телевизора, автоматическое переключение телевизора в режим работы с компьютером при подключении соединительного кабеля. Необходимо сразу отметить, что для соответствующей доработки телевизора потребуются только два широко распространенных гнезда типа СГ-5 и, возможно, три резистора. Набор дополнительных радиодеталей, устанавливаемых в БК, и того меньше — один пятиконтактный разъем, подобный разъемам БП, МГ, ТВ компьютера, и один резистор.

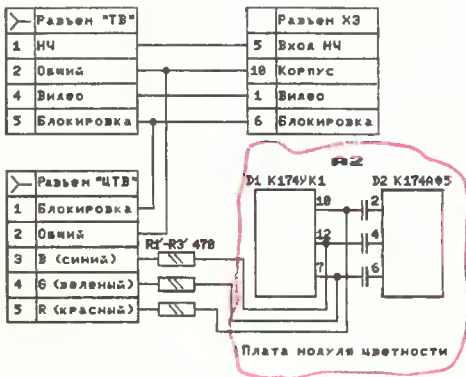
Схемы распиновки двух розеток типа СГ-5, устанавливаемых в телевизоре, а также соединительного кабеля приведены на рис. 1 и 2 соответственно. Контакты разъема ТВ с помощью проводников соединяются с контактами разъема ХЗ, установленного на плате модуля радиоканала А1 (по схеме телевизора). При этом необходимо удостовериться в наличии в цепи «Блокировка» всех обозна-



ченных на схеме телевизора соединений и радиоэлементов. В частности, на плате А1 должна быть установлена перемычка М-М, а на плате submodule радиоканала А1.3 элементы VD1, VD2, R34. Резисторы R1'—R3', показанные на рис. 1, в случае наличия на плате модуля цветности А2 резисторов аналогового назначения (например, резисторов R84—R86 для модуля цветности МЦ-2) устанавливать не нужно. Степень насыщенности на экране телевизора основных цветов компьютерной графики можно регулировать, изменяя номиналы этих сопротивлений в пределах 330—680 Ом.

Разъемы ТВ, ЦТВ очень удобно установить на передней панели телевизора вместо установленных там двух разъемов, предназначенных для подключения телефонов и магнитофона. В результате такой переделки дизайн телевизора останется неизменным. Поскольку для подключения магнитофона в телевизоре использована розетка типа СГ-5, ее можно не снимать, а достаточно просто перепаять в соответствии с рис. 1.

При доработке БК-0010 необходимо разъем компьютера, ХТ9 (ТВ) перепаять согласно схеме распиновки разъема ТВ телевизора. Для этого контакты 1 и 5 разъема ХТ9 сначала необходимо освободить от всех старых связей. Затем его контакт 1 нужно соединить с контактом 5 разъема ХТ4 (МГ). Дополнительный разъем ХТ11 (ЦТВ), устанавливаемый в компьютере рядом с разъемом ХТ9, распаивается в соответствии со схемой, показанной на рис. 3. Дорожку, связывающую вывод 13 микросхем D20 (по схеме БК-0010) с выводом 5 микросхем D21, необходимо перерезать. Соединение выхода элемента D20.4 с выводом 5 микро-

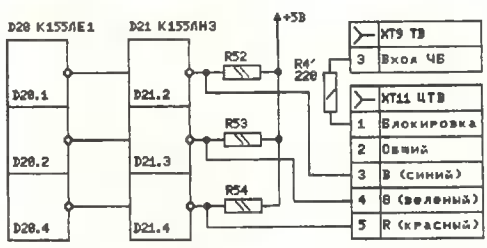


схемы D21 выполняется отрезком тонкого проводника навесным способом.

Отличительная особенность рассматриваемой схемы состоит не только в том, что для своей реализации она требует минимум дополнительных радиоэлементов, но также и в том, что ее выходные R, G, B-сигналы являются инвертированными по отношению к сигналам цветности, которые получаются на выходе схемы, опубликованной в журнале «Информатика и образование» (1989, № 2). Необходимость инвертирования сигналов цветности может возникнуть и при использовании других типов цветных телевизоров.

В режиме работы с цветом на контакт 4 разъема ТВ компьютера должен подаваться сигнал «Синхронизация», а при работе с черно-белым изображением — сигнал «Видео». Для получения требуемой коммутации этих двух сигналов между контактом 3 разъема ТВ и выводами элементов C9, R55, ранее соединявшихся с контактом 5 этого же разъема, должны быть включены либо нормально замкнутые (при работе с черно-белым изображением) контакты механического переключателя, либо такие же контакты реле, срабатывающего от +5 В. В последнем случае один вывод катушки реле через нагрузочный резистор подключается к источнику +5 В, а другой — к контакту «Блокировка» разъема ЦТВ компьютера.

Альтернативным способом получения синхронизирующего сигнала для цветного изображения является использование в качестве такого сигнала «Видео». Однако при таком решении для достижения устойчивой синхронизации и правильной цветопередачи изображения, возможно, придется подобрать номинал резистора R4' (а возможно, и совсем его исключить), включенного между контактом 3 разъема ХТ9 и контактом 1 разъема ХТ11



(см. рис. 3). Кроме того, на плате БК между правыми выводами (по схеме компьютера) элементов C9, R55 и левым выводом резистора R56 необходимо установить перемычку.

Подключение БК к телевизору для работы с цветом осуществляется с помощью соединительного кабеля, показанного на рис. 2. Для перехода в режим черно-белого изображения достаточно просто вынуть штекер ЦТВ из одноименного разъема компьютера (и перевести в соответствующее положение переключатель, если он используется). Автоматическое переключение телевизора из режима работы с компьютером в обычный режим показа телевизионных передач происходит при отключении от разъемов ТВ и ЦТВ телевизора обоих штекеров соединительного кабеля.

Правильность описанной методики доработки компьютера БК-0010 для работы с цветной графикой и подключения его к телевизору была практически проверена на нескольких цветных телевизорах с торговыми индексами Ц-275, Ц-280. Проверка показала, что выполнение всех требуемых операций по распайке соединительного кабеля, доработке БК и телевизора займет у радиолюбителя средней квалификации не более одного свободного вечера.

П. ЧИРКОВ

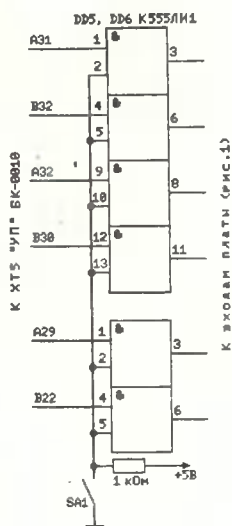
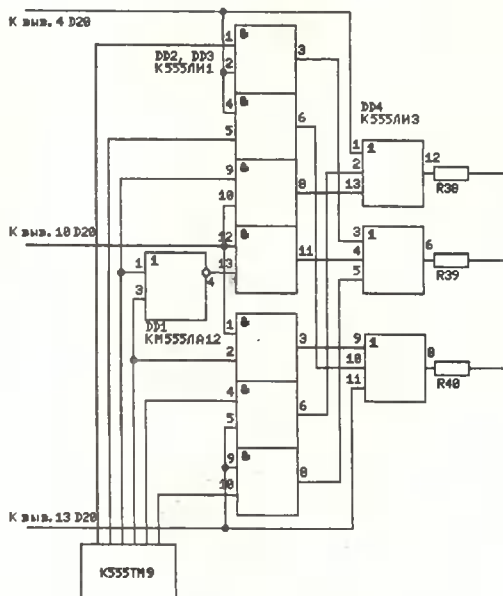
70

## Еще раз о восьмицветном БК

В статье «Радуга на экране БК» (1989, № 2) был описан способ расширения цветовых возможностей БК. Наиболее удобным для пользования является, конечно, метод доработки схемы. Но предложенный В. Кондрашовым способ увеличения количества воспроизводимых цветов на экране довольно сложен и связан с необратимыми изменениями в конструкции компьютера. Предлагаю вариант, который избавляет от необходимости зашивки в ПЗУ, значительно упрощает процесс переделки и не вносит изменений в конструкцию БК. Возможности воспроизведения цветов почти такие же (63 различные палитры).

Первая часть переделки для получения трех цветов остается без изменений. После удачного ее осуществления необходимо отпаять концы резисторов R38, R39, R40 от дорожек, ведущих к микросхеме D20, и в получившиеся разрывы впаять выводы платы, схема которой приведена на рис. 1. Эта схема представляет собой управляемый коммутатор, который вместе с элементами формирователя цветов компьютера заменяет ПЗУ K556РТ4, предлагаемую В. Кондрашовым. При исправных микросхемах никакой настройки не требуется. Палитры здесь фиксированные, но их количество и разнообразие вполне удовлетворяют любого вла-





дельца БК. Еще больше упростить схему можно путем замены регистра K555ТМ9 регистром выходного порта. При этом выводы порта подпаиваются непосредственно к входам дополнительной платы. Можно использовать те же разряды (9—14), только код палитры необходимо записывать по адресу 177714<sub>8</sub>. Неудобство данного изменения в том, что при работе с внешними устройствами через данный порт будет происходить смена цветов. Поэтому можно предусмотреть отключение управления дополнительными цветами, как показано на рис. 2.

Таблицу соответствия управляющих чисел и палитр можно составить, используя логику схемы, но помещать в журнале нет необходимости, поскольку после изготовления устройства можно ввести в компьютер элементарную программу, рисующую на экране трехцветный рисунок и автоматически изменяющую число (от 177000<sub>8</sub> и до 100000<sub>8</sub> с интервалом 1000<sub>8</sub>), заносимое в

управляющий порт (177716<sub>8</sub> или 177714<sub>8</sub>).

Владельцам БК с Фокалом следует воспользоваться функцией обращения к общей шине. При этом числа заносятся в десятичной системе: —512, —1024, —1536 и т. д., уменьшая каждый раз на 512 до —32768.

Указанные микросхемы можно заменить аналогичными. Лучше использовать серии 555, 133, поскольку они меньше нагружают блок питания компьютера.

В заключение несколько слов о применении полученных эффектов. Видимо, читатели поняли, что одновременно на экране дисплея всегда будет присутствовать только три цвета (кроме черного), но программное переключение палитр позволит создать изображение, переливающееся различными цветами.

А. КУЗНЕЦОВ,  
учитель-методист  
п. Кумёны  
Кировской обл.

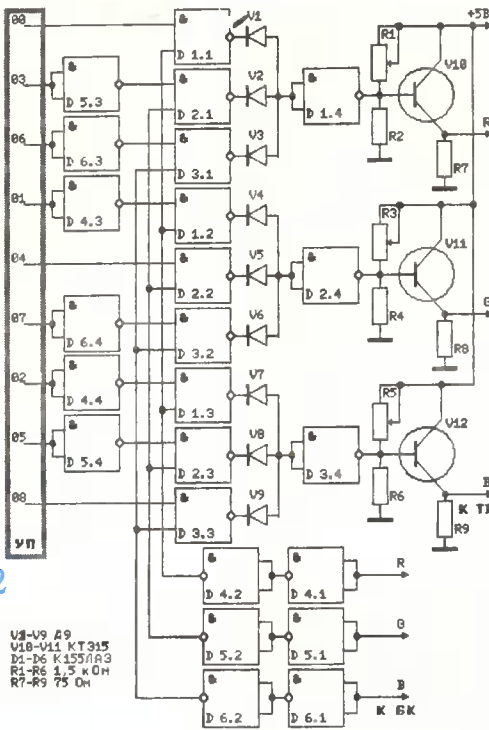
## Восьмицветная приставка

Получить восемь цветов на экране БК можно и без внесения необратимых изменений в его конструкцию. Это позволяет сделать описываемая ниже приставка. Ее использование позволяет также программировать палитру, т. е. в любой момент изменить ее.

Впрочем, от недостатков она не свободна — занимает порт УП БК, так что программы, работающие, например, с принтером, могут отображаться только в стандартных цветах. Джойстик и другие устройства можно под-

ключить к этой приставке, если они используются не более шести контактов BD (байта данных) — контакты A16, A13, B12, B10, B5, B7, B6, A7 УП).

Элементы D4.3—4.4, D5.3 — 5.4, D6.3—6.4 приставки служат для обеспечения стандартного цвета при включении питания и после сброса; элементы D4.1—4.2, D5.1.—5.2, D6.1—6.2 увеличивают нагрузочную способность БК и формируют уровни TTL выходов цвета; элементы D1.1—1.3, D2.1—2.3, D3.1—



V1-V9 Д9  
V10-V12 КТ315  
D1-D6 К1551А3  
R1-R6 1,5 кОм  
R7-R9 75 Ом

Расположение битов цвета  
в порте "УП" (адрес 177714)

08	07	06	05	04	03	02	01	00
BB	BG	BR	GB	GG	GR	RB	RG	RR
B			G			R		

3.3 — дешифратор кода палитры; V1—V9 — диодное «ИЛИ», служат также для развязки элементов дешифратора (можно использовать любые маломощные импульсные диоды); D1.4, D2.4, D3.4 приводят сигнал цвета к стандартной полярности; R1—R9 и V10—V12 служат для регулировки уровней цветовых сигналов.

После включения питания или сброса все биты порта УП равны 1. Благодаря инверторам D4.3, 4.4, 5.3, 5.4, 6.3, 6.4 элементы

Редакции журнала хотелось бы знать мнения читателей о конструкциях, описания которых публикуются в «Клубе БК». Сообщите нам: какая конструкция вас заинтересовала? Можете ли вы самостоятельно собрать ее или предпочли бы купить готовую? Описания каких конструкций вы хотели бы прочесть в журнале?

На конверте с ответами сделайте пометку «Хардвер БК».

Таблица дополнительных цветов  
и соответствующие им коды

Код	B	G	R
0	синий	зеленый	красный
1	черный	черный	черный
2	фиолетовый	желтый	желтый
3	красный	красный	зеленый
4	голубой	голубой	фиолетовый
5	зеленый	синий	синий
6	белый	белый	белый
7	желтый	фиолетовый	голубой

D2.1, 3.1, 1.2, 3.2, 1.3, 2.3 будут заперты и красный цвет пойдет только на красную «пушку» телевизора, зеленый — на зеленую, синий — на синюю. Палитра экрана будет стандартной.

Если же в порт УП будут занесены нули, цвет экрана изменится. Покажем это на примере.

Чтобы вместо красного цвета получить желтый, следует по адресу 177714 записать код «02». Бит BD01 установится в 0, инвертор D4.3 разрешит прохождение сигнала красного цвета и через элемент D1.2, т. е. на зеленую «пушку». Смешение красного и зеленого цветов даст желтый.

Этот способ обеспечивает восемь цветов. Для обогащения палитры можно через оставшиеся разряды порта УП управлять яркостью цветов.

Обращение к палитре осуществляется командой MOV # код, # порт, т. е. занимает три слова.

Переназначив основные цвета на один основной или дополнительный, можно на цветном мониторе просматривать тексты в режиме 64 символа в строке. Код черно-белого текста, например, 666.

Лучше всего смотрятся программы, написанные с учетом применения данной приставки, но и уже существующие программы легко раскрасить в более естественные цвета.

С. СМЕРНОВ

От редакции. Переделка заметно повышает нагрузку на блок питания БК, поэтому стоит, быть может, сделать для приставки собственный источник питания.

*«Прошу откликнуться владельцев игры «Бой в памяти» для БК-0010».*

*В. В. НОСОВ, 680052, Хабировск,  
ул. Жуковского, 3, кв. 1.*

*«При подключении БК-0010 к телевизору «Электроника Ц401М» возникают затруднения. С цветами все в порядке, но синхронизация неустойчива».*

**И. СТУПАЛЬСКИЙ**

Для повышения устойчивости синхронизации на плате видеоусилителя телевизора нужно переключить перемычку Ш-2 на контакт

2—3 (обозначения даны в соответствии с принципиальной схемой телевизора). Лучше всего сделать это с помощью двухпозиционного переключателя — тогда можно будет менять режим работы телевизора: просмотр программ/работа с ПЭВМ.

*«Интересуюсь совмещением БК с каналом MIDI в цифровых музыкальных синтезаторах. Обменяюсь программами для БК (имею более 100 различных программ)».*

*К. К. БОРИСЕНКО, 250032, Чернигов,  
ул. Пухова, 103, кв. 88*

## **Вниманию руководителей организаций и учебных заведений, использующих в своей работе компьютеры БК-0010!**

Школьный информационный кооператив «ШИК» предлагает программно-методический комплекс по обучению работе на клавиатуре БК-0010 слепым десятипальцевым методом (методика О. В. Ивановой, см.: Информатика и образование. 1988. № 3). Курс рассчитан на 12 занятий по 45 мин, причем зрительный образ клавиатуры формируется уже на втором занятии.

Знание клавиатуры БК-0010 и умение работать «вслепую» не только повысит производительность труда ваших сотрудников и учащихся, но и уменьшит время их пребывания перед экраном компьютера.

**Наш адрес:** 121069, Москва, ул. Б. Молчановка, 26/28, школа № 1234, К. А. Шульману.

**Наши телефоны:** 291-85-73, 141-43-13.

## **«СОЗИДАНИЕ»**

Универсальный кооператив предлагает гражданам и организациям программное обеспечение и схемы дополнительных устройств для БК-0010, БК-0010.01.

С авторами заключаются договоры на тиражирование их разработок. Желающих установить деловые связи с кооперативом «Созидание» просим обращаться по адресу:

**480100 Алма-Ата, пр. Ленина, д. 30, кв. 11, Шторк И. В.  
Тел.: 61-22-68 (код Алма-Аты: 327-0).**

# ЧТО МОЖЕТ ЭВМ

## Сквозняк против радиации

В пользу свежего воздуха убеждать никого не надо. Однако многие, особенно зимой, достаточно часто проветривают свой кабинет. А ведь в воздухе закупоренного помещения помимо других вредных химических и механических компонент накапливается невидимый и не имеющий запаха радиоактивный газ радон, выделяемый строительными конструкциями зданий.

Интенсивность выделения радона непостоянна. Она зависит от материалов, из которых построено здание, времени года, атмосферного давления и может быть разной в соседних квартирах и даже комнатах. Поэтому точно определить момент, когда количество газа превысило допустимую норму, можно только с помощью прибора, который выпускается одной из американских фирм.

Основной прибор является микропроцессор, который обрабатывает поступающую с датчика информацию и накапливает ее. Если радиоактивность воздуха в помещении менее 4 пикокюри на литр, на лицевой панели прибора горит зеленая лампочка. При превышении допустимой нормы начинает мигать красный индикатор. Кроме того, прибор может «припомнить» максимальный и минимальный уровни радиоактивности воздуха в помещении за последние сутки, а также ее интегральное значение за прошедшие 12 часов и даже за весь год.

А пока вы не приобрели такой прибор, старайтесь почаще проветривать помещение.

## Размером с... батарею

Эти слова могли до недавнего времени послужить отличной рекламой любому электронному прибору, сегодня в гонке за компактностью электроника значительно опередила миниатюризацию источников энергии: все чаще и чаще размер, форму и даже сферу применения миниатюрных электро-

ных устройств определяют элементы питания.

Выйти из тупика помогут новые батарейки, разработанные совместно двумя японскими фирмами, которыми был изобретен сухой, похожий на листок бумаги гибкий электродит. На основе такого накопителя энергии можно изготавливать батарейки, которые можно гнуть, скручивать и даже мять, придавая им необходимую форму.

Такие источники питания не только позволяют резко уменьшить размеры миниатюрных приборов, в которых они используются, но и дадут возможность создавать совершенно необычные пока устройства, например такие, как полупроводниковый элемент со встроенным питанием или линия связи, улучшающая передаваемый по ней сигнал.

## Музыка души

Если у вас нет слуха, вы не владеете ни одним из музыкальных инструментов, не можете выразить словами звучащую у вас в душе музыку — вам поможет новый электронный прибор, разработанный двумя молодыми американцами: инженером и музыкантом.

Прибор, которому авторы дали недвусмысленное название «Биомьюз», преобразует биотоки мозга, мускулов рук, ног и век, поступающие с датчиков, в музыку, воспроизводимую синтезатором. Программа, управляющая работой оригинального музыкального инструмента, написана таким образом, чтобы как можно точнее преобразовать жестиколярные и мимические ассоциации человека в музыкальную форму. Например, направление взгляда перемещает источник звука в стереофоническом пространстве, работа мускулов ног задает музыкальный ритм, движением кистей формируется гармонический ряд, а биотоки мозга определяют мелодию. Программа может быть настроена под конкретного исполнителя, что позволит ему наиболее полно реализовать свои творческие возможности.

Закрепив датчики, «биомузыкант» выходит на сцену и начинает изящно пританцовывать, жестиковать, мечтать, и весь зал может слышать, о чем поет его душа. И хотя человек, имеющий классическую музыкальную подготовку, возможно, сумеет извлечь

из нового инструмента значительно более изящную мелодию, «Биомьюз», по мнению авторов, принесет огромную пользу и людям с поврежденными функциями моторного контроля, а также тем, кто надолго прикован к постели. «Биомьюз» — это еще и средство «музыкальной терапии», и способ почувствовать себя полноценным человеком, способным творить и создавать прекрасное.

## Мой дом — моя слабость

Жилье для современного человека — это не только возможность укрыться от непогоды, отдохнуть или принять гостей. Дом — это еще и место приложения наших неистощенных сил. Именно поэтому всё, что может сделать жизнь человека удобнее и приятнее, в первую очередь появляется в жилище. Сегодня на наши квартиры идет походом электроника: часы и электронные игры, программируемые стиральные комплексы и телевизоры, бытовые компьютеры и электронные пишущие машинки. Чтобы как-то упорядочить и укротить этот разнородный поток электроники, национальная ассоциация жилищного строительства США в 1984 г. создала проект под названием «Удобное жилище», что было смелой попыткой скоординировать развитие домашней автоматизации.

К 1988 г. были сформулированы основные требования, и в Балтиморе было начато строительство первых «домов будущего». С внешним миром такой дом связан проволочным кабелем, по которому поступают электрический ток для питания, телефонный, радио-, видео-, а также высокоскоростной цифровой информационный сигнал. Все приборы автоматического «дома будущего», включающие батальон микропроцессоров, связаны между собой и с внешними компьютерами большей мощности. Развешенная электроника обеспечит полную безопасность дома, удобство в управлении им, ненавязчивость и своевременность услуг.

И хотя члены инициативной группы ассоциации понимают, что стандарт может устареть раньше, чем станет реальностью, общественность с интересом относится к их проекту, в котором сфокусированы идеи, призванные изменить стиль нашей жизни в новом тысячелетии.

Л. ШТЕРНБЕРГ

Куйбышевский авиационный институт

## Уроки с программируемыми микрокалькуляторами

### Подпрограммы

После изучения темы «Вспомогательные алгоритмы» можно перейти к реализации их на ПМК. На первом уроке рекомендуется дать команды «ПП» и «В/О» без упоминания стека возвратов: просто отметить, что команда «ПП» запоминает адрес возврата в некотором регистре, содержимое которого увидеть невозможно, а «В/О» выполняет переход по запомненному адресу. Объяснение работы этих команд займет не более 10 мин. Сами по себе команды «ПП» и «В/О» позволяют организовывать только подпрограммы без параметров. Такие подпрограммы либо всегда выдают один и тот же результат, либо должны работать с переменными (всегда с одними и теми же), принадлежащими основной (головной) программе.

В этом месте учитель встретится с одной методической тонкостью. В учебнике нет ни явного запрета, ни явного разрешения на использование во вспомогательных алгоритмах значений переменных из основного алгоритма, хотя неявно предполагается, что связь идет только по параметрам, а все переменные, описанные после *нач*, — *локальные* (собственные) переменные вспомогательного алгоритма: при исполнении вспомогательного алгоритма мы рисуем таблицу со своими столбцами для каждой такой переменной.

В языках программирования эта проблема решена по-разному: в Фортране, например, доступа из подпрограммы к переменной из головной программы нет, в Паскале — есть, в

Бейсике — только так и идет работа и т. д. Учителю можно порекомендовать такой вариант: если переменная не описана ни среди параметров, ни после *нач*, то предполагается, что вспомогательный алгоритм берет переменную, описанную в основном алгоритме.

В качестве примера подпрограммы без параметров можно взять подпрограмму набора физической или химической константы, известной школьникам, например числа Авогадро ( $6.02252 \cdot 10^{23}$ ). Для ее набора нужно 10 команд: «6.0 2 2 5 2 ВП 2 З». Предположим, что она нужна в расчете более одного раза, а для ее хранения регистра не нашлось (все заняты другими данными). Тогда выделение этой последовательности команд в подпрограмму сократит программу, позволив не выписывать всю последовательность многократно.

Для иллюстрации работы команд «ПП» и «В/О» можно ввести рассмотренную выше программу в память ПМК с адреса 20, а с адреса 00 простейшую головную программу; «ПП 20 2—ПП 20» и пройти ее в пошаговом режиме (класс работает при этом в «командном режиме») с заглядыванием в счетчик адреса до и после выполнения команд «ПП» и «В/О» (табл. 1).

И так далее. После второго обращения к подпрограмме убедимся, что будет выполнен возврат «под» вторую команду «ПП».

Одна из задач написания вспомогательных алгоритмов — это «обучение» исполнителя новым последовательностям действий, которые становятся для него новыми командами (см. [1]). Этот момент следует акцентировать и проиллюстрировать на примере: после того как в памяти ПМК оказалась упомянутая выше подпрограмма, мы

75

Таблица 1

Клавиши	Индикатор	Комментарий
F ПРГ	00	Стоим перед выполнением команды «ПП».
F АВТ ПП	0. 0.	Выполнили команду «ПП».
F ПРГ	00 00 00	20 Убедились, что выполнен переход на адрес 20.
F АВТ ПП, ПП	0. (10 раз)	Проходим по шагам подпрограмму.
F ПРГ	0С 02 03	30 Стоим перед командой «В/О».
F АВТ ПП	6.02252 6.02252	23 23 Выполняем команду «В/О».
F ПРГ	20 53	02 Убеждаемся, что вернулись «под» команду «ПП».
F АВТ ...	6.02252 ...	23

можем считать, что ПМК «обучен» новой командой — команда «ПП 20» мало чем теперь отличается от команды «Fл» и выполняет (как и команда «Fл») занесение на регистр X некоей специфической константы, хотя и делает это гораздо дольше, чем занесение л.

Реализацию подпрограмм с параметрами можно выполнить на ПМК в точном соответствии с тем, как это описано в учебнике [2]. Для выполнения вспомогательного алгоритма заводится своя таблица переменных, значения аргументов заполняются значениями фактических параметров, выполняется тело, затем значения результатов переписываются из таблицы вспомогательного алгоритма в соответствующие переменные из таблицы основного алгоритма. Следовательно, на ПМК для переменных, используемых в подпрограмме, надо выделить регистры, отличные от тех, которые используются в головной программе; в регистры, соответствующие аргументам, записать значения фактических параметров, обратиться к подпрограмме, а затем из регистров, соответствующих результатам, переслать значения в регистры, отведенные переменным головной программы. Проиллюстрируем это на примере подпрограммы, выполняющей деление нацело с остатком двух чисел по алгоритму:

алг *нацело* (цел  $a, b$ , *часть*, *ост*)

— арг  $a, b$ ; рез *часть*, *ост*

нач *часть* :=  $[a/b]$ ;

ост :=  $a - b \cdot \text{часть}$

КОН

к которому мы обращаемся командой *нацело* ( $aa - bbb, s, \text{ча}$ , *ост*)

Для используемых в головной программе переменных  $aa, bbb, s, \text{ча}$  и *ост* отведем регистры P0 — P4 соответственно, для используемых в подпрограмме переменных  $a, b, \text{ча}$  и *ост* — P5—P8. Обратите внимание (как свое, так и учеников), что у нас имеются две переменные *ост*: своя — у головного алгоритма и своя — у вспомогательного (табл. 2).

Этой реализацией обращений к подпрограмме можно воспользоваться в слабых классах, однако она страдает двумя недостатками: во-первых, неэффективностью, во-

Таблица 2

Адрес	Команда	А-язык	Комментарий
00	П—х 0	<i>нацело</i> (	Значение разности $aa$ и $bbb$ пересылаем в регистр, соответствующий первому аргументу подпрограммы, значение второго аргумента — в регистр, соответствующий второму аргументу; выполняем собственно обращение к подпрограмме (адрес ее размещения пока выбран произвольно), а теперь пересылаем результаты из регистров, выделенных подпрограмме, в регистры, выделенные головной программе
01	П—х 1		
02	—	$aa - bbb,$	
03	х—П 5		
04	П—х 2		
05	х—П 6	$s,$	
06	ПП		
07	20		
08	П—х 7	$\text{ча},$	
09	х—П 3		
10	П—х 8	<i>ост</i> )	
11	х—П 4		
...	...	...	
20	П—х 5	<i>часть</i> :=	С адреса 20 начинается подпрограмма, которая работает только с отведенными ей регистрами, не заботясь о том, откуда взялись аргументы и куда денутся результаты. Команды вычисления остатка выписывать не будем
21	П—х 6		
22	÷		
23*	F [ ]	$[a/b]$	
24	х—П 7		
25	...		
...	...		
31	V/O		

\* Для ПМК БЗ-34, МК-54, где нет команды «F [ ]», для выделения целой части из заведомо положительного числа надо выполнить последовательность команд «х—П  $m$ , К П—х  $m$ , П—х  $m$ », где  $m$  — один из номеров регистров из диапазона 7 — D.

вторых, методической неточностью. Такая передача параметров предполагает, что головная программа «знает», где что размещается в подпрограмме, а вот это как раз она «знать» и не обязана.

Отметим (этот момент нужно акцентировать и на уроке), что для использования вспомогательного алгоритма достаточно знать его заголовок, а тело может быть написано как угодно. Например, если в приведенной выше подпрограмме и в соответствующем ей алгоритме заменить тело на цикл последовательного вычитания:

```

част:=0; ост:=a;
пока ост>b нц
  ост:=ост - b; част:=част + 1
кц

```

то обращаться к алгоритму (подпрограмме) можно точно так же, как и ранее.

Но и заголовок нужно знать не весь. Поясним это на примере заголовка алгоритма суммирования таблицы, содержащего параметры разных типов (чем он для наших целей и удобен):

**алг сумма (вещ таб a [1:M], цел M, вещ сум)**

**арг a, M; рез сум;**

Перепишем его в несколько иной форме, объединив список параметров и списки аргументов и результатов (методически очень удачная форма, используемая в [4], которую будем применять далее):

**алг сумма (арг вещ таб a [1:M], цел M, рез вещ сум)**

Теперь, если убрать отсюда имена переменных, то останется именно то, что надо знать для использования алгоритма:

**алг сумма (арг вещ таб, цел, рез вещ)**

а именно: алгоритму надо дать два аргумента — вещественную таблицу и целое число, а выдает алгоритм один вещественный результат. При этом то, как называются эти параметры внутри вспомогательного алгоритма, знать нет необходимости. При таком изложении становится очевидным, что параметры сопоставляются по порядку следования, а не по именам.

Итак, вспомогательному алгоритму (подпрограмме) надо выдать аргументы в нужном порядке, а затем забрать результаты. Наиболее естественно это сделать через стек. Применительно к рассмотренной выше подпрограмме деления с остатком получим следующее (табл. 3).

Отметим, что если бы наша подпрограмма выдавала только один результат, например остаток, то ее работа шла бы внешне по тому же принципу, что и любой двуместной операции: выбираем операнды из стека и оставляем в стеке результат, т. е., написав

Адрес	Команда	А-язык	Комментарий
00	П—х 0	наце- ло (	
01	П—х 1		
02	—	aa —	Заносим в стек значение 1-го аргумента, заносим в стек значение 2-го аргумента, выполняем собственно обращение к подпрограмме;
03	П—х 2	c,	
04	ПП		
05	20		
06	х—П 3	ча,	
07	F o		теперь забираем из стека оставленные там подпрограммой результаты
08	х—П 4	ост);	
...	...		
20	х—П 6		Подпрограмма выбирает из стека аргументы и засылает их туда, куда ей надо.
21	↔		
22	х—П 5		либо обрабатывает непосредственно в стеке.
23	÷	част:=	
...	...	...	Результаты оставляет в стеке в нужном порядке
...	V/O		

подпрограмму, мы как бы научили ПМК еще одной операции: «остаток».

Разобрав на уроке эти (или аналогичные примеры), следует сформулировать правила программирования вспомогательных алгоритмов.

1. Значения аргументов вызова вспомогательного алгоритма засылаются в стек в порядке их перечисления в списке параметров.

2. Ставится команда «ПП» с пока незаполненным адресом начала подпрограммы.

3. Значения результатов извлекаются из стека в порядке их перечисления в списке параметров.

4. По завершении программирования основной программы определяется адрес, с которого размещается подпрограмма (очередной свободный адрес), — этим адресом заполняются пустые адреса команд «ПП».

5. Для параметров и прочих используемых в подпрограмме переменных вводятся регистры из числа не занятых в головной программе и ранее написанных подпрограммах.

6. Подпрограмма начинает работу с извлечения из стека значений парамет-

ров в порядке, обратном порядку их перечисления в списке аргументов.

7. Тело подпрограммы программируется как обычно.

8. Результаты засылаются в стек в порядке, обратном порядку их перечисления в списке результатов.

9. кон для подпрограммы программируется командой «В/О».

Добавим к этим правилам несколько комментариев для учителя, связанных с возможными вопросами учеников.

1. Что делать, если аргументов или результатов более четырех?

Для школьных алгоритмов это ограничение не актуально, но в целом это вопрос того же типа, что и «А если программа не помещается в памяти ПМК?»: у ПМК довольно ограниченные возможности, и далеко не все на нем реализуется.

78 2. Как передать через параметры таблицу?

Малая память ПМК делает почти невозможным размещение в ней более одной таблицы, поэтому передача таблиц через параметры мало актуальна, но вопрос интересен и учитель должен знать ответ на него. Тот способ передачи параметров, который описан в учебнике и который рассмотрен здесь, носит название «передача параметров по значению-результату» и в практических языках программирования используется нечасто. Чаще используется другой прием — «передача параметров ссылкой» (применяется в Паскале, Фортране, ПЛ/1 и др.), когда подпрограмме передается адрес переменной или таблицы, а не копия ее значения. Использование этого приема на ПМК описано в [3], с. 111—113, но обсуждать его, по-видимому, целесообразно только на кружках, в математических школах или очень сильных классах.

3. Некоторые особенности ПМК заставляют отводить память для используемых в подпрограммах переменных одновременно с отведением памяти в головной программе: для используемых в подпрограмме циклов и косвенных переменных могут потребоваться регистры из числа первых.

В качестве домашнего задания следует дать одну-две задачи на реализацию на ПМК вспомогательных алгоритмов из числа разобранных ранее на уроке. Очень показательны такие задачи:

**Задача 1.** Написать вспомогательный алгоритм, вычисляющий по длинам сторон треугольника  $a$ ,  $b$ ,  $c$  угол, лежащий против стороны  $a$ , и тремя обращениями к нему вычислить все углы треугольника.

Сам алгоритм очень прост:

алг угол (арг вещ  $a$ ,  $b$ ,  $c$ , рез вещ  $uz$ )

нач  $uz := \arccos((b^2 + c^2 - a^2) / (2 \cdot b \cdot c))$

кон

а три обращения к нему отличаются только порядком задания аргументов

угол ( $a$ ,  $b$ ,  $c$ ,  $uzA$ ); угол ( $b$ ,  $c$ ,  $a$ ,  $uzB$ );

угол ( $c$ ,  $a$ ,  $b$ ,  $uzC$ )

при этом получаются разные результаты.

Этот пример прекрасно иллюстрирует сопоставление формальных и фактических параметров по порядку их следования в списках, а не по именам переменных: после того как значения аргументов засланы в стек, у подпрограммы нет никаких средств определить, откуда взялось значение, находящееся в том или ином регистре стека. Программу для слегка усложненного варианта этой задачи можно найти в [3], с. 107—110.

**Задача 2.** Написать вспомогательный алгоритм, вычисляющий длину отрезка по координатам вершин, и головной алгоритм, который тремя обращениями к вспомогательному получает длины сторон треугольника, координаты вершин которого заданы в таблицах  $X[1:3]$  и  $Y[1:3]$ , где  $(X[K], Y[K])$  — координата одной из вершин. Реализовать алгоритмы на ПМК.

Алгоритмы имеют вид:

алг стороны (арг вещ таб  $X[1:3], Y[1, 3]$ , рез вещ  $a12, a23, a31$ )

нач длина ( $X[1], Y[1], X[2], Y[2], a12$ )

длина ( $X[2], Y[2], X[3], Y[3], a23$ )

длина ( $X[3], Y[3], X[1], Y[1], a31$ )

кон

алг длина арг вещ ( $x1, y1, x2, y2$ , рез вещ дл)

нач  $дл := \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$

кон

Их реализация на ПМК проблем не вызывает. При их обсуждении на следующем уроке нужно обратить внимание на то, что аргументами вспомогательного алгоритма являются вещественные переменные, а фактическими параметрами — элементы таблицы. Это допустимо, так как элемент вещественной таблицы — вещественный, а после засылки аргументов в стек никаким способом нельзя определить из каких переменных — простых или элементов таблиц — взяты значения.

После анализа домашнего задания можно уточнить работу команд «ПП» и «В/О», введя понятие стека возвратов, который позволяет из одной подпрограммы обращаться к другой. В качестве примера, где такое обращение необходимо, можно взять вспомогательный алгоритм вычисления площади криволинейной трапеции или уточнения корня методом деления пополам, которые обращаются к алгоритму вычисления функции. Соответствующие программы можно найти в [3], с. 101—105, 114—117.



Полезно отметить, что, в отличие от описанного в учебнике, на ПМК головной и вспомогательный алгоритмы отличаются тем, что кон для головного реализуется командой «С/П», а для вспомогательного — «В/О», а также по-разному передаются параметры. Но если бы команда «В/О» при пустом стеке возвратов выполняла останов (а не переход на адрес 01), то подпрограмма не отличалась бы от головной программы. Что же касается параметров, то и для головной программы их можно задавать в стеке, а не в регистрах (такой прием называется СТ-вводом — см. [3], с. 123). Иными словами все несоответствия описанному в учебнике — это мелкие детали, вызванные особенностями используемой техники.

Следующий «калькуляторный» урок посвящается реализации на ПМК функций, но предварительно разберем предшествующий ему урок, на котором вводится это понятие, так как мотивация его введения в учебнике недостаточна.

Начнем с определения понятия функции и назначения отдельных частей ее описания.

*Функция (или, более строго, алгоритм вычисления значения функции) — это способ оформления вспомогательного алгоритма, имеющего единственный результат, который не является табличной величиной.*

Например, рассмотренный выше алгоритм нацело нельзя оформить как функцию, так как он выдает два результата; алгоритм, заполняющий таблицу нулями, не может быть функцией, так как его результат хоть и единственный, но табличный; а алгоритм суммирования таблицы можно оформить как функцию: он выдает единственный нетабличный результат — сумму.

Функции удобнее в использовании, так как позволяют не заводить лишних переменных и сокращают запись. Например, если у нас имеется вспомогательный алго-

ритм определения большего из двух чисел в варианте нефункции и в варианте функции с заготовками соответственно:

алг БИД (арг вещь  $a, b$ , рез вещь  $m$ )  
и

алг вещь бид (вещь  $a, b$ )

то для вычисления максимума из трех нам придется писать

либо БИД ( $a, b, m$ ); БИД ( $m, c, mm$ );

либо  $mm := бид(бид(a, b), c)$ .

Это и является основным достоинством функций, но, к сожалению, не каждый алгоритм может быть оформлен как функция.

Поскольку у функции единственный параметр-результат, то роль этого параметра берет на себя имя функции. Описатель типа, стоящий перед именем функции, относится как раз к этому имени (а также к переменной с именем знач) и показывает, какого типа значения может принимать это имя. Поскольку имя функции взяло на себя роль параметра-результата, то в списке параметров остается на один параметр меньше, а так как все оставшиеся параметры — аргументы, то нет необходимости в списке аргументов: и так ясно, что все параметры являются аргументами (хотя авторы [4] и оставили слово арг в функциях, но в нем нет необходимости). Так как имя функции получает значение, то обращение к функции можно использовать в выражениях на правах простой переменной. Итак, чтобы превратить алгоритм в функцию, надо:

а) описатель типа результата поставить перед именем алгоритма;

б) убрать переменную-результат из списка параметров, а в теле алгоритма заменить ее всюду на знач,

в) убрать список аргументов и список результатов (или слово арг, если изложение дается по [4]) (см. рис. 1 а, б, в).

Собственно реализация функций на ПМК не вызывает никаких проблем: функции реализуются по общим правилам подпрограмм,

а) алг БИД (арг вещь  $a, b$ , рез вещь  $m$ )  
арг  $a, b$ ; рез  $m$   
кон если  $a > b$  то  $m := a$   
иначе  $m := b$   
кон

б) алг вещь БИД (вещь  $a, b, m$ )  
арг  $a, b$ ; рез  $m$   
кон если  $a > b$  то  $m := a$   
иначе  $m := b$   
кон

в) алг вещь БИД (вещь  $a, b$ )  
арг  $a, b$ ; рез  $m$   
кон если  $a > b$  то знач :=  $a$   
иначе знач :=  $b$   
кон

единственный результат при засылке в стек оказывается в регистре X и может быть использован не только для засылки куда-то, но и непосредственно в арифметических операциях в точном соответствии с тем, как обычно используют результат функций. Однако особенности ПМК накладывают некоторые ограничения на использование функций:

а) при засылке в стек аргументов многоаргументных функций можно вытолкнуть из стека какой-либо полезный промежуточный результат;

б) для уверенного программирования выражений, содержащих обращения к функциям, необходимо, чтобы функции *корректно использовали стек*, т. е. не оставляли в нем ничего лишнего (подробнее об этом термине см. [3], с. 117).

Чтобы не усложнять материал этими деталями, проще потребовать, чтобы обращение к функции было всегда первым членом выражения.

В качестве задач для рассмотрения на уроке и домашних заданий годятся почти все задачи учебника, в которых использованы функции.

Финальным аккордом темы «Вспомогательные алгоритмы» целесообразно провести урок по схеме «цепочка задач», иллюстрирующей, как облегчает работу наличие библиотеки подпрограмм. В качестве цепочки задач можно взять, например, такие (см. [2], часть 2, с. 52):

1) определить а) последнюю, б) предпоследнюю, в) К-ю цифру целого числа;

2) определить сумму цифр трехзначного числа;

3) определить, является ли «счастливым» билет с данным номером (можно написать и алгоритм подсчета числа «счастливых»

билетов, но он невыполним за разумное время на ПМК).

Задачи 1а и 1б — вводные. Общий случай дает задача 1в. С использованием рассмотренного алгоритма *нацело* алгоритм *цифраК* имеет вид

алг цел *цифраК* (цел число, к);

нач цел *част*, *ост*;

*нацело* (число,  $10^k$ , *част*, *ост*);

*нацело* (*ост*,  $10^{k-1}$ , знач, *ост*);

кон

Теперь, имея алгоритм *цифраК*, легко получить алгоритм *суммацифр*:

нач знач := *цифраК* (число, 1) + *цифраК*

(число, 2) + *цифраК* (число, 3)

Итак, в нашей библиотеке алгоритмов есть и алгоритм *нацело* и алгоритм *суммацифр*. С их помощью проверка «счастливости» билета выполняется очень просто: *нацело* (номер, 1000, циф123, циф456); если *суммацифр* (циф123)-*суммацифр* (циф456) то...

Программы получаются не самые оптимальные по скорости, но отлично показывают, как «набирается» алгоритм из крупных блоков вместо его построения из кирпичиков-команд.

Рассмотрев эти алгоритмы в классе и записав в виде программ для ПМК в домашних заданиях, на последнем уроке вводим алгоритм *нацело* в дальние адреса памяти (скажем, с адреса 80), а с адреса, например, 60 пишем головную программу — алгоритм *цифраК*. Убедившись, что все работает, заменяем «С/П» в программе *цифраК* на «В/О», превращая ее в подпрограмму, и пишем с адреса 40 программу *суммацифр* и т. д. (Заметим только, что  $10^k$  необходимо вычислять циклом умножения, ибо использование команд «F  $10^k$ » дает большую погрешность.)

#### Литература

1. Шень А. Информатика в 9 классе // Информатика и образование. 1987. № 6. С. 5—9.
2. Основы информатики и вычислительной техники / Под ред. А. П. Ершова и В. М. Монахова. М.: Просвещение, 1985.
3. Штернберг Л. Ф. Программирование на микрокалькуляторе. М.: Просвещение, 1988.
4. Основы информатики и вычислительной техники / Под ред. А. П. Ершова. М.: Просвещение, 1988.

*От редакции.* Уважаемые читатели! Этой статьей мы заканчиваем цикл «Уроки с ПМК». Автору и редакции хотелось бы получить от вас отзывы на эти публикации: понравились ли они вам, пригодились ли в работе, нужно ли в дальнейшем освещать эту тематику на страницах журнала? Ждем ваших писем!



## Компьютерные игры и орфография

Коммуникация между человеком и компьютером происходит на уровне письменных сообщений. Практика показывает, что 90 % ошибок, которые допускают учащиеся при работе с компьютером, вызваны неправильным набором слов на клавиатуре, т. е. связаны с орфографией. Компьютер, анализируя письменные ответы, как правило, сравнивает их с эталонами правильных ответов, заложенных в программу, и если одна или несколько букв ответа не совпадают с эталоном, то ЭВМ сигнализирует об ошибке. При этом для компьютера безразлично, допущена ли серьезная грамматическая ошибка или же ошибкой является неправильно написанная буква.

Это, с одной стороны, тормозит работу учащегося, но, с другой стороны, способствует формированию у него навыков грамотного письма. Наблюдения показывают, что после занятий в дисплейном классе учащиеся начинают серьезнее относиться к орфографии. Например, иностранные студенты во время традиционных занятий очень часто пишут и вместо й (причем преподавателю иногда бывает трудно убедить студентов в том, что это разные буквы). Компьютер же в этом случае, неустанно сигнализируя об ошибке, очень быстро добивается необходимого результата. Студент на практике убеждается, что неправильное написание отдельных букв создает реальные помехи для коммуникации, отсюда — более внимательное отношение к орфографии.

Таким образом, компьютер является прекрасным тренажером по выработке навыков правописания. В связи с этим мы не можем согласиться с некоторыми разработчиками компьютерных программ по русскому и иностранным языкам, которые предлагают заменить ввод письменного ответа на изучаемом языке обыкновенными цифровыми кодами, мотивируя это различными причинами. Одни, например, полагают, что в тех случаях, когда контролируется не правописание, а, скажем, семантика [1], можно пользоваться цифровым кодом ответов, другие разработчики используют коды в целях экономии времени и т. п.

Однако и те и другие забывают о том, что обучение языку должно происходить на комплексной основе. Нельзя развивать одни навыки в ущерб другим и, конечно, говоря о ликвидации компьютерной неграмотности (а разработчики компьютерных программ так или иначе участвуют в этом процессе), ни

в коем случае не следует способствовать развитию у учащихся обыкновенной, элементарной неграмотности.

На наш взгляд, способ цифрового кодирования ответов, возникший еще в ту пору, когда возможности ЭВМ были существенно ограничены и не допускали буквенного ввода ответа, на настоящем этапе обучения языкам с помощью компьютера допустим в незначительном объеме лишь на самом начальном этапе изучения иностранного языка в течение первой-второй недели.

Хорошие результаты при обучении правописанию на компьютере дают зрительные диктанты. Суть их заключается в том, что то или иное слово появляется на экране дисплея на короткое время, достаточное для его прочтения, а затем исчезает. Задача учащегося — набрать данное слово на клавиатуре. И если это сделано верно, то компьютер предъясвляет следующую единицу зрительного диктанта и т. д. Если же допущена ошибка, то слово выдается на экран дисплея еще раз, время показа несколько увеличивается.

На начальном этапе обучения, когда навыки грамотного письма на русском или иностранном языке еще только формируются, зрительные диктанты должны предшествовать любой компьютерной программе. Таким образом с их помощью заранее снимаются орфографические трудности, поэтому учащиеся при выполнении основной программы, следующей сразу же за зрительным диктантом, могут сосредоточить внимание на выполнении конкретной задачи.

Неплохие результаты дают и компьютерные программы на подстановку пропущенных букв, например:

— Вставьте пропущенные буквы: **МОЛ...КО.** В подобных программах возможен как немедленный, так и отсроченный контроль. В первом случае учащиеся сразу же видят ответную реакцию компьютера (**—Правильно! МОЛОКО.**) или неправильно (**—Ошибка! Правильный ответ: МОЛОКО.**) В случае же отсроченного контроля компьютер реагирует несколько иначе лишь на ошибочный ответ:

— **Ошибка!** Подумайте еще и дайте правильный ответ. Образцовый ответ выдается учащемуся лишь после двух неправильных ответов: — Вы ошиблись уже два раза. Правильный ответ: **МОЛОКО.**

По нашим наблюдениям, интерес к подобным программам во много раз возрастает, если ввести в программу даже незначитель-

ный игровой элемент. И это вполне понятно, так как хорошо известно, что можно превратить любую однообразную и неинтересную работу в игру, если поставить игровые по форме цели, которые совпадают с рабочими задачами [4]. Люди поняли это еще несколько веков назад, достаточно вспомнить замечание Д. С. Лихачева о том, что «потеха и учение у Петра I все время объединялись, вопреки поговорке, их противопоставлявшей: «Ученью время — потехе час» [3].

Нужно отметить, что разработчик компьютерных программ по русскому или иностранному языку при выборе игрового элемента имеет перед собой огромное поле деятельности. Причем очень часто нет необходимости изобретать игровые элементы, нужно лишь повнимательнее взглянуть глазами лингвиста, глазами преподавателя на уже существующие игровые компьютерные программы. Оказывается, что даже самые «бесполезные» с точки зрения обучения русскому языку игровые программы можно видоизменить так, что они становятся реальными помощниками в изучении различных аспектов языка.

Вспомним, к примеру, известную динамическую компьютерную игру «Тетрис», в которой играющий должен аккуратно укладывать в сосуде тетрамино фигурки из четырех квадратов, падающие по экрану сверху вниз на кладку из этих фигурок. Задача игрока — так заполнить сосуд, чтобы не оставалось «окошек» в кладке из тетрамино.

Приглядевшись к этой игре, мы и создали свой вариант этой игры, получившей название «Орфограф». Роль тетрамино в нем играют слова с пропущенными буквами, а в нижней части сосуда располагаются отдельные буквы (рис. 1). Задача игрока — опустить слово так, чтобы пропуск в слове попал на нужную букву. Можно создать и обратный вариант, когда учащийся должен опустить отдельную букву в то или иное слово с пропуском. Так, казалось бы, бесполезная с точки зрения обучения языку игра

1



помогает выработке у учащихся навыков правописания.

Конечно, из сказанного совсем не следует, что разработчик компьютерной программы по языку должен брать за основу только существующие игровые программы. Здесь, как мы уже отмечали, благодатное поле и для личного творчества. Назовем, к примеру, разработанные нами игровые программы: «Восхождение на пик Орфографии», «Прыгун», «Орфоохота», которые предназначены для автоматизации навыков правописания.

«Восхождение на пик Орфографии» начинается с подготовки, заключающейся в том, что обучаемому предъявляется ряд слов изучаемого языка, правописание которых необходимо запомнить для успешного восхождения. После того как обучаемый подготовится, компьютер выводит на экран изображение горы с десятью уступами. Каждый уступ — это новое препятствие, которое необходимо будет преодолеть учащемуся. У подножия горы появляется изображение человека.

Основная работа с программой начинается тогда, когда на первом уступе загорается слово с пропущенной буквой, например: ГОР...Д. Учащийся должен ввести в компьютер недостающую букву. Если ответ правильный, то человек на экране поднимается на первый уступ, а на втором уступе загорается новое слово. В случае неправильного ответа компьютер сообщает об этом учащемуся и предлагает попытаться ответить еще раз. Если и на этот раз попытка не приводит к успеху, то компьютер сообщает «альпинисту», что ему необходимо еще раз пройти инструктаж, после чего возвращает учащегося к исходному списку слов. Повторив слова, обучаемый может снова начать восхождение.

Если учащийся успешно поднялся на первый или любой другой уступ, но при подъеме на следующий ошибся, то он автоматически скатывается на один уступ вниз и вынужден повторять восхождение. Когда учащийся добирается до шестого уступа, то попадает в зону тумана (на экране верхние уступы закрывает облако). Здесь характер работы изменяется, вместо слов с пропусками в условиях «ограниченной видимости» появляется целое слово, но лишь на короткое время, затем туман снова заволакивает слово. Компьютер сообщает учащемуся, что ему требуется написать все слово (зрительный диктант). Правильный ответ поднимает обучаемого еще на один уступ, в случае же неправильного ответа «туман» снова проясняется, но уже на большее время. Таким образом, «альпинист», не скатываясь вниз, получает возможность ответить еще раз. Если же и вторая попытка оказывается неудачной, то «альпинист» ска-

тывается сразу на два уступа.

Учащегося, достигшего вершины, компьютер поздравляет и сообщает ему время, которое обучаемый затратил на все восхождение. Если это время меньше времени, затраченного на аналогичное восхождение преподавателем русского или иностранного языка, то компьютер обязательно поздравляет играющего с новым мировым рекордом.

В тех случаях, когда программа «Восхождение на пик Орфографии» используется автономно, ее лексическое наполнение может каждый раз меняться с помощью датчика случайных чисел из определенного банка слов с орфограммами.

Если же подобная программа призвана снять орфографические трудности для работы с другой (основной) программой, то лексическое наполнение первой заранее устанавливается разработчиком и включает все наиболее трудные в орфографическом плане слова основной программы.

В программе «Прыгун» (рис. 2) обучаемый должен вставлять пропущенные буквы в слова, появляющиеся под планкой. Если буква определена верно, то спортсмен преодолевает высоту, в противном случае он сбивает планку. С каждым удачным прыжком планка поднимается на одно деление вверх. Неудачный прыжок, наоборот, опускает планку на деление вниз. При достижении максимальной высоты игроком компьютер объявляет его победителем, а при минимальном уровне планки обучаемый выбывает из соревнований.

В игровой программе «Орфоохота» (рис. 3) обучаемый должен «выстрелить» правильной



3

буквой по пролетающей орфоптице (слово с пропуском). Если выстрел удачен, то птица падает. «Охотник» имеет возможность стрелять по одной и той же цели дважды. Компьютер, подсчитывая промахи и добытых птиц, выдает обучаемому игровые результаты в конце игры. Движение птиц по экрану осуществляется в трех скоростных режимах (медленном, среднем и быстром), выбор которых производится перед началом игры самим учащимся.

83

Таким образом, включая игровую компоненту в программы по развитию навыков правописания, мы реально подтверждаем мнение о том, что «компьютерные игры и процесс обучения с помощью компьютеров, постепенно сближаясь, видимо, скоро окажутся единым и нерасторжимым понятием» [2].

## Литература

1. Власов Е. А. Опыт создания компьютерной программы обучения лексической синонимии // Русский язык за рубежом. 1988. № 2. С. 63—66.
2. Дьюдни А. К. Путешествие с роботами в Роботрополе и овладение навыками конструирования электронных микросхем // В мире науки. 1985. № 9. С. 94—99.
3. Лихачев Д. С. Поэзия садов: К семантике садово-парковых стилей. Л.: Наука, 1982.
4. Шмелев А. Г. Мир поправимых ошибок // Компьютерные игры: Обучение и психологическая разгрузка // Новое в жизни, науке, технике. Сер. «Вычислительная техника и ее применение». М.: Знание. 1988. № 3.

2



## ПЭВМ в помощь учителю химии

Школьный учитель, прикованный к столу написанием практических, контрольных, экзаменационных работ, бессчетное число раз выписывает ученикам задания, умножая их на виды и варианты. При подготовке к уроку он должен каждый раз рыться в книжных и журнальных полках, в многочисленных тетрадках, папках и выписках, снова и снова искать аргументы и материалы, которыми можно будет воспользоваться на уроке.

Используя возможности компьютера, у него появляется шанс если не разорубить гордиев узел, то заметно его ослабить.

Конкретно эта идея преломилась в комплексе программ для автоматизированного рабочего места учителя (АРМ «УЧИТЕЛЬ») который реализован на персональной ЭВМ «Роботрон РС-1715» в операционной системе SCP посредством СУБД «DBASE-2»

Объектом автоматизации выбран учебный материал курса неорганической химии VII класса. С помощью учителя-предметника в память компьютера заложена информация по разделам «Начальные химические понятия», «Химическая связь», «Периодический закон», «Галогены» и др.

Вся информационная база программно структурирована, что позволяет выделить в ней разделы, темы и подтемы, принятые в учебном плане. При проведении занятий учитель использует разнообразный методический материал, раскрывающий пункты плана. Для этого программно предусмотрено ведение информационной базы по таким видам учебно-методического материала, используемым при изучении данного курса, как *задачи, экспериментальные задачи, лабораторные работы, упражнения, вопросы, литература к теме*. Кроме того, в память компьютера закладываются проблемные и парадоксальные ситуации, которые ученики будут разбирать по ходу урока, материалы к различным демонстрационным опытам.

При проектировании информационной базы предусмотрено наличие вариантов по видам учебного материала и градация его по трем степеням сложности: низшей, средней и высшей, чтобы избежать традиционных цифр — «приговоров». Такой подход позволяет каждому ученику выбрать вариант, который ему по силам, а при успешном его решении взять более сложный билет и попытаться повысить свою оценку. Все это экономит время как учеников, так и учителя, делает процесс общения на уроке более демократичным и увлекательным.

Раздел: Начальные химические понятия  
Тема: Вычисление количества вещества по схеме реакции

Задача

Сложность: высшая  
Вариант: 1544304

В евриметре смешали 8 г кислорода и 1 г водорода. После этого смесь подожгли. Сколько воды образовалось?

Раздел: Начальные химические понятия  
Тема: Простые и сложные элементы

Лабораторная работа

Сложность: низшая  
Вариант: 1137105

Тема: Определение среды

Цель: С помощью индикаторов определить среду некоторых моющих средств

*Материалы и оборудование:* раствор фенолфталеина, метилоранжа, растворы моющих средств, зола, фильтровальная бумага, чашка Петри, пипетка

*Порядок и техника работы:*

*Моющие средства:* Фенолфталеин : Метилоранж : Выводы

Раздел: Начальные химические понятия  
Тема: Простые и сложные элементы

Экспериментальная задача

Сложность: низшая  
Вариант: 1134104

Разделить смесь железных опилок и растительного масла.

*Оборудование:* воронка, фильтровальная бумага, магнит, сосуд с водой

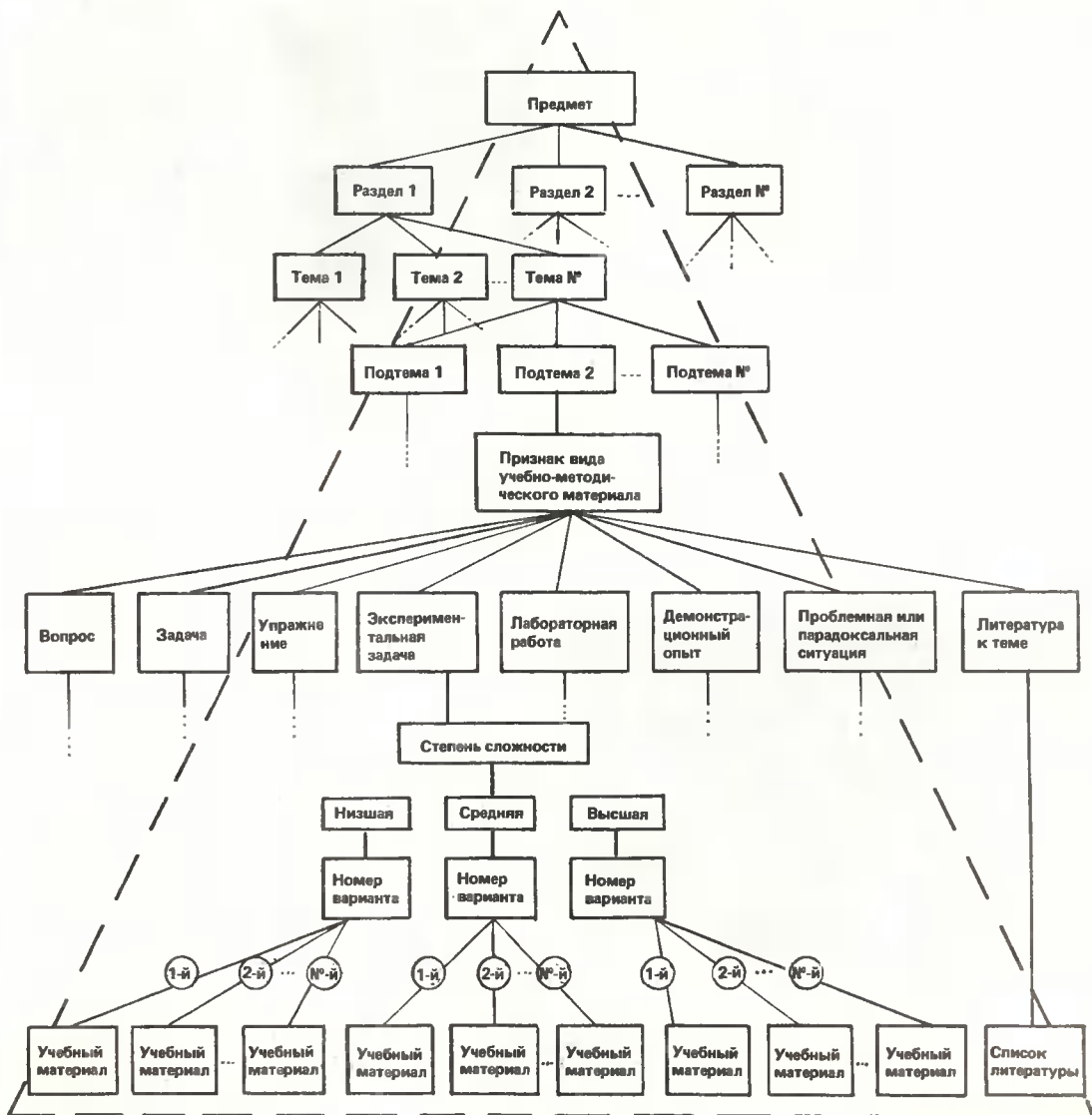
Базу можно представить в виде пирамидальной схемы, изображенной на рис. 1. Верхними «этажами» обозначаются изучаемые разделы школьного курса. Нижестоящий уровень («этаж») подчиненности разбивается на входящие в раздел темы; те, в свою очередь, на подтемы; каждая из подтем — на виды учебно-методического материала, далее — на степени сложности и варианты. Между «этажами» существуют структурные связи; самый нижний «этаж» — основание пирамиды — представляет собой «кирпичики» информационной базы, т. е. учебный материал.

Благодаря программам поиска и представления информации заложенная в компьютер информационная база превращается в авто-

матерIALIZED библиотеку, архив учителя, где по запросу выдается нужный учебный материал. Выглядит это так: учитель садится за компьютер, на экране дисплея появляется инструкция по работе с АРМ «УЧИТЕЛЬ». Ознакомившись с ней, он «переворачивает» следующую страницу, где записаны режим работы АРМа, выбирает нужный и продолжает диалог дальше. Например, чтобы из памяти машины выбрать экспериментальную задачу низшей степени сложности, в основном «меню» необходимо указать режим выдачи методического материала и с появлением следующего рабочего «меню» последовательно задать кодами изучаемый раздел — «начальные химические понятия», тему — «атомно-молеку-

лярное учение», подтему — «простые и сложные вещества», вид учебно-методического материала — «экспериментальная задача», степень сложности — «низшая» и требуемый вариант. При неправильном задании кодов выдается сообщение с указанием ошибки и предоставляется возможность коррекции. В результате на печать будет выдан билет-задание. Коды, используемые при выборке учебного материала, отражают структурные характеристики информационной базы, описаны в инструкции-памятке и быстро запоминаются. Последовательность их задания указывает путь от вершины пирамиды к ее основанию — выбранному «кирпичику» информационной базы.

По желанию учитель может вызвать и рас-



печатать весь имеющийся материал по теме урока в нужном количестве экземпляров. Им может быть рекомендованная литература, вопросы к теме, упражнения и т. д.

Описанная система является открытой. Каждый учитель может ее дополнить или внести свои коррективы. Для этого ему представлена возможность ввода, отображе-

ния и корректировки имеющейся информации.

С помощью компьютера каждодневные усилия учителя по поиску, сбору и обработке материалов не окажутся бесследными, а «зацементируются» в память ЭВМ. Так компьютер помогает тем, кто учит, и тем, кого учат.

Е. ЛАЗАРЕВА

## Помогает центр информатики

В условиях острой нехватки компьютеров трудно рассчитывать на успешное изучение основ информатики в средней школе. Опыт, накопленный в Москве, свидетельствует, что изменить сложившуюся ситуацию способны центры, создаваемые по инициативе Государственного комитета СССР по вычислительной технике и информатике при активной помощи подведомственных ему учреждений, партийных и советских органов.

В Тушинском районе, например, заботу по организации такого центра взяли на себя коллективы Всесоюзного научно-исследовательского института межотраслевой информации (ВИМИ) и других ведущих предприятий и организаций. Немало пришлось потрудиться, занимаясь перепланировкой и капитальным ремонтом помещения, наладкой выделенной вычислительной техники, подготовкой специалистов для работы по разным направлениям.

Сегодня помимо подготовки работников



местных предприятий, на которых планируется внедрение компьютеров, и переподготовки специалистов, выполнения расчетов на договорных началах и разработки программ существенное внимание уделяется здесь обучению основам информатики и вычислительной техники учеников 882-й (подшефной) и еще трех средних школ района. Учебные занятия школьников на персональных ЭВМ «Агат» и «Корвет», размещенных в специально отведенном зале, ведутся бесплатно, а их факультативная и досуговая деятельность — за счет средств (достаточно скромных), перечисляемых школьными родительскими комитетами. По свидетельству учителей, использование материальной базы центра и помощь его специалистов в разработке обучающих программ позволяют успешно приобщать ребят к компьютерной культуре и способствуют лучшему усвоению других учебных предметов. Довольны и сами школьники, и их родители.



КСК

Поправки к № 6—89 г.  
На стр. 63, правый столбец, верхняя строка, вместо  $D > 1$  должно быть  $D > 9$ .

На с. 113 формулу следует читать так:

$$\sum_{i=1}^s (n_i^r - m^r)$$



С. КУЗНЕЦОВ, В. РАСПОПОВ

## Формирование навыков параметрического программирования

Навыки параметрического программирования, т. е. умение учащегося воспользоваться готовым программным обеспечением ЭВМ для решения конкретной задачи, проявляются в умении квалифицированно выбрать те или иные значения параметров задачи и ввести их в программу. Рассматриваемые в курсе ОИВТ алгоритмы и программы численного интегрирования, численного решения уравнений, нахождения экстремумов безусловно способствуют формированию у учащихся стиля параметрического программирования. Но все эти алгоритмы относятся к сфере вычислительной математики, тогда как современные компьютеры большую часть времени заняты решением задач логического анализа и преобразования информации. Поэтому в дополнение к рассматриваемым в курсе задачам невычислительного типа (задачи сортировки элементов таблиц, обработки литературных величин) на внеклассных занятиях целесообразно рассмотреть задачи анализа логических игр, которые вызывают у учащихся повышенный интерес.

Изоморфизм рассматриваемых в статье логических игр, симметрия соответствующих им игровых полей позволили сформулировать задания, которые формируют у учащихся навыки параметрического программирования. Суть заданий — параметрическая настройка игровой программы, выполнить которую можно, лишь детально разобравшись в выигрышном алгоритме игры.

Задание 1. Игра «Крестики-нолики» знакома всем учащимся. Конструктивным доказательством существования беспроигрышного алгоритма в ней может служить программа для микрокалькулятора или персонального компьютера.

Чтобы побудить кружковцев к анализу

игры и поиску беспроигрышного алгоритма, учитель предлагает учащимся сразиться с компьютером, используя игровое поле с перенумерованными клетками (рис. 1). Машина вступает в игру первой и никогда не проигрывает. В этом кружковцы очень скоро убеждаются.

7	8	9
4	5	6
1	2	3

— Программа устроена таким образом, — комментирует учитель, — что компьютер первым ходом занимает центральное поле 5. Допустим, учащийся сделал ответный ход на поле 1 и ввел этот номер клетки игрового поля в ЭВМ. В ответ он получает очередной ход компьютера, и т. д. Если учащийся играет невнимательно, то он проигрывает. При внимательной игре соперника компьютер после своего четвертого хода сводит партию к ничейному исходу. А если после первого хода компьютера «5» вы ответите, например, ходом на боковую клетку «2», то ЭВМ обязательно выиграет: своим третьим ходом она делает «вилку», нейтрализовать которую вы уже не в состоянии.

Затем учитель предлагает кружковцам убедиться в том, что ЭВМ «думает» именно по этому алгоритму, сыграв с ней еще несколько партий.

— Теперь попробуйте записать беспроигрышный алгоритм, — ставит задание учитель.

Вскоре учащиеся понимают сущность алгоритма, по которому действует компьютер:

**алг** БЕСПРОИГРЫШНЫЙ АЛГОРИТМ КОМПЬЮТЕРА (нат  $X_0, A, B, C, D, E$ )

нач хожу на  $X_0$ ; отвечаю ходом на А, угрожая В;  
если угроза не ликвидируется, то хожу на В и выигрываю,  
иначе хожу на С, угрожая ходом на D;  
если угроза не ликвидирована, то хожу на D и выигрываю,  
иначе если первый ход противника был в угловую клетку,  
то ход в Е обеспечивает ничью, иначе хожу на Е и выигрываю;

все

кон  
 Величины  $X_0, A, B, C, D, E$  — это параметры алгоритма; конкретные значения величин А, В, С, D, Е зависят от выбора значения  $X_0$ , обозначающего номер поля, на которое компьютер делает первый ход.

88 При разборе текста игровой программы учитель обращает внимание учащихся на то, что в ней использованы массивы пятизначных чисел — ABCDE. Величины А, В, С, D, Е — это значения соответствующих разрядов элементов массива  $X(1), X(2), \dots, X(9)$ . Для случая, когда компьютер делает первый ход на центральную клетку 5, т. е. при  $X_0=5$ , массиву  $X(1), X(2), \dots, X(9)$  присваиваются значения в соответствии с табл. 1. При этом развитие партии определяется первым ответным ходом человека.

После анализа этой игровой программы учитель предлагает учащимся 8 вариантов однотипных заданий по параметрическому программированию: перепрограммировать компьютер таким образом, чтобы он делал первый ход в клетку, отличную от центральной. При выполнении каждого из этих вось-

ми заданий учащимся предстоит логически проанализировать возможные продолжения партий при различных значениях параметра  $X_0=\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$  и представить результаты анализа в виде таблицы, аналогичной табл. 1. Затем осуществляется параметрическая настройка программы на новые условия игры.

Целесообразно перед выдачей этого задания акцентировать внимание учащихся на следующих обстоятельствах:

а) задача во всех восьми случаях, имеет решение;

б) при первом ходе компьютера в угловую клетку возможен лишь один ответный ход человека, приводящий к ничейному исходу партии, при первом ходе компьютера в боковую клетку таких ответных ходов четыре;

в) найденные пятизначные коды продолжений партии следует представить в виде таблицы, аналогичной табл. 1; параметрическая настройка игровой программы на новые условия игры заключается в том, что пятизначные коды продолжений партий заносятся в соответствующие ячейки ПМК либо присваиваются соответствующим элементам массива  $X$  в программе на Бейсике, при этом тело программы не изменяется.

Дидактическое достоинство рассмотренной выше игровой программы «Крестики-нолики» заключается в том, что она допускает параметрическую настройку и на случай других игр, которые изоморфны игре «Крестики-нолики». Продемонстрируем это.

Задание 2. Игра «Слова» ([1], с. 222).

На каждой из девяти карточек написано по одному слову:

1. БАНЯ
2. БУСЫ
3. ГАТЬ
4. МАРС
5. ОРЕХ
6. ПЛУГ
7. СНЕГ
8. УРОН
9. ХЛЕБ

Двое игроков поочередно берут по карточке. Выигрывает тот, кому первому удастся взять три карточки, имеющие общую букву в словах.

Изоморфизм игр «Слова» и «Крестики-нолики» станет очевидным, если расположить занумерованные карточки со словами в виде таблицы, представленной на рис. 2.

— В качестве игровой программы воспользуйтесь ранее рассмотренной программой «Крестики-нолики», выполнив ее параметрическую настройку на игру «Слова», — предлагает следующее задание кружковцам учитель. — Как и в предыдущем случае,

Таблица 1

Ячейки адресуемой памяти ПМК	Обозначения элементов таблицы	Значения величин					
		а	Е	D	С	В	А
0	$X_0$	5					
1	$X_1$	—	6	3	3	7	8
2	$X_2$		6	1	9	7	3
3	$X_3$	—	4	1	9	8	2
4	$X_4$		2	7	3	9	1
5	$X_5$	0					
6	$X_6$		2	7	3	9	1
7	$X_7$	—	4	1	9	8	2
8	$X_8$		6	1	9	7	3
9	$X_9$	—	6	3	7	8	2

Знак «минус» в столбце а указывает на те продолжения партии, которые завершаются ничейным исходом.

2

	Б	Г	Р
С			
У	2 БУСЫ	6 ПЛУГ	8 УРОН
Е	9 ХЛЕБ	7 СНЕГ	5 ОРЕХ
А	1 БАНЯ	3 ГАТЬ	4 МАРС
Н			

3

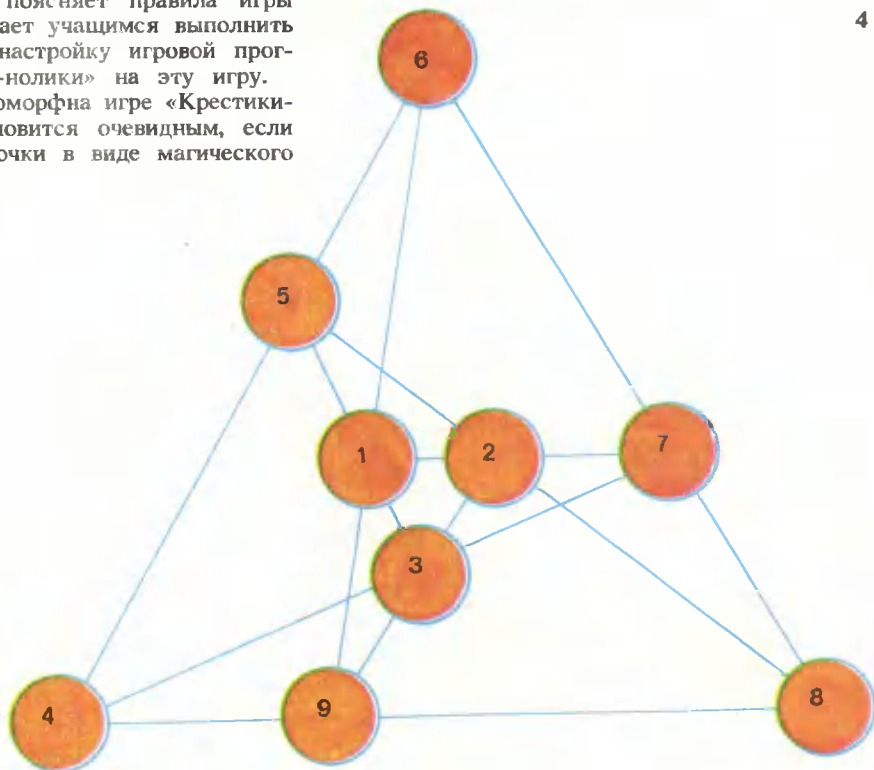
	15	15	15
15			
15	4	9	2
15	3	5	7
15	8	1	6

диалог человека и ЭВМ ведется с указанием номеров карточек, которые берут соперники.

Задание 3. Игра «15 на новый лад» ([1], с. 215).

— Имеются 9 карточек с числами от 1 до 9. Два соперника поочередно берут себе по карточке. Выигрывает тот, кому первому удастся набрать на трех взятых карточках сумму, равную 15,— поясняет правила игры учитель и предлагает учащимся выполнить параметрическую настройку игровой программы «Крестики-нолики» на эту игру.

Данная игра изоморфна игре «Крестики-нолики», что становится очевидным, если расположить карточки в виде магического квадрата (рис. 3).



Задание 4. Игра «Тригекс» ([2], с. 307—314).

Каждый из двух игроков запасается четырьмя фишками. Делая ходы поочередно, игроки выставляют фишки на кружки игрового поля, изображенного на рис. 4. Выигравшим считается тот, кто первым сумеет занять три кружка, расположенные вдоль одной прямой.

Отметим, что, хотя поле для игры в «Тригекс» не изоморфно полю для игры в «Крестики-нолики», тем не менее можно использовать ранее рассмотренные алгоритм и программы для игры «Крестики-нолики» (в данном случае имеет место изоморфизм алгоритмов и программ).

Перед демонстрацией этой игровой программы на компьютере необходимо выполнить ее параметрическую настройку в соответствии с табл. 2.

— В игре «Тригекс» компьютер начинает партию и всегда выигрывает. Попробуйте проанализировать развитие игры для различных вариантов первого хода компьютера и выполните соответствующую каждому случаю параметрическую настройку этой игровой программы,— формулирует очередное задание кружковцам учитель.

Задание 5. Эту игру, изоморфную игре «Тригекс», предложила ученица IX класса

89

4

Таблица 2

Ячейки адресуемой памяти ПМК	Обозначения элементов таблицы	Значения величин					
		а	Е	D	С	В	А
0	X <sub>0</sub>	9					
1	X <sub>1</sub>		5	3	2	4	8
2	X <sub>2</sub>		7	1	6	4	8
3	X <sub>3</sub>		5	1	6	8	4
4	X <sub>4</sub>		5	6	1	2	3
5	X <sub>5</sub>		7	6	1	3	2
6	X <sub>6</sub>		5	4	8	3	2
7	X <sub>7</sub>		5	6	1	2	3
8	X <sub>8</sub>		7	6	1	3	2
9	X <sub>9</sub>	0					

кировоградской средней школы № 34 Оля Вдовенко. Правила те же, что и в игре «Слова», но набор слов на карточках иной:

1. АИСТ
2. УТРО
3. ПЕСО
4. ХЛЕБ
5. СРУБ
6. БАНЯ
7. ТЕНЬ
8. НУЛЬ
9. ЛОЗА

Изоморфизм этой игры игре «Тригекс» иллюстрирует рис. 5.

Игровая программа для ПМК  
"Электроника БЗ-34"

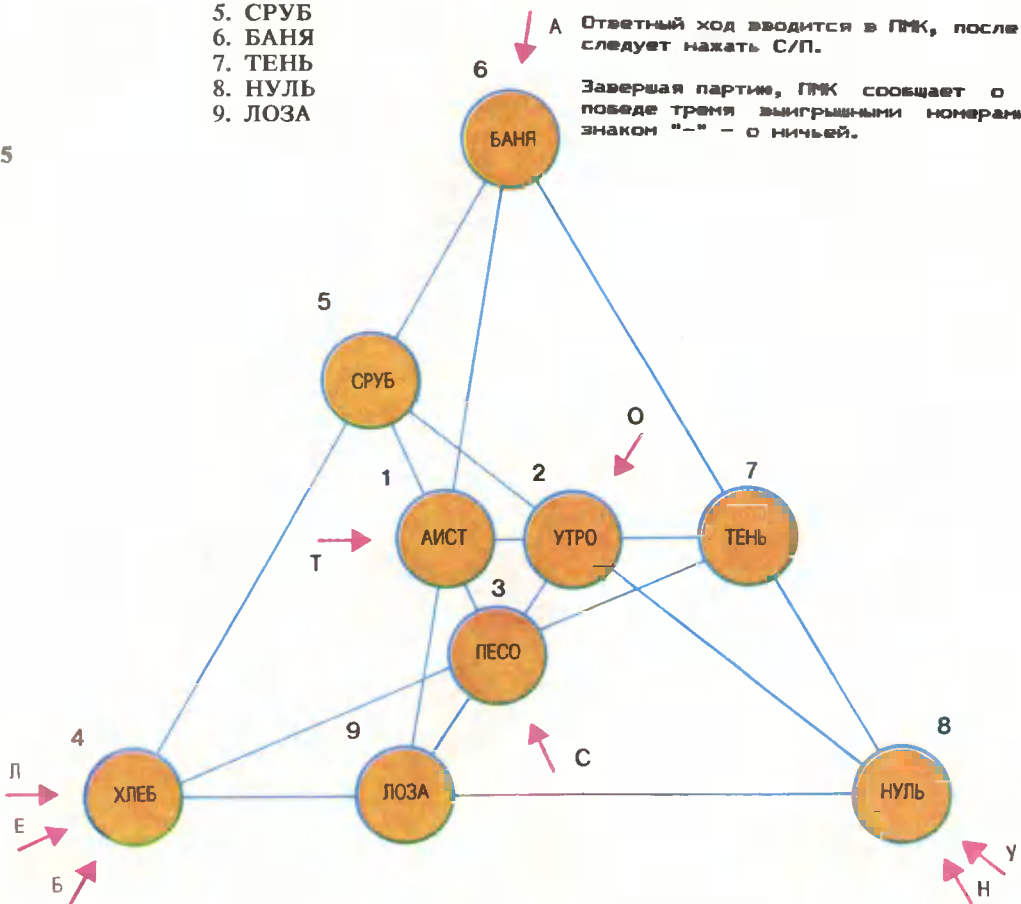
00.ИПО	01.С/П	02.ПД	03.КИПД	04.ПС
05.Фх<0	06.08	07./-/	08.ПД	09.ПП
10.60	11.ПА	12.С/П	13.ПП	14.60
15.-	16.Фх#0	17.23	18.ФВх	19.ИПО
20.ИПА	21.БП	22.47	23.ПП	24.60
25.ПВ	26.С/П	27.ПП	28.59	29.-
30.Фх#0	31.37	32.ФВх	33.ИПО	34.ИПВ
35.БП	36.47	37.ИПС	38.Фх<0	39.44
40.ИПД	41./-/	42.БП	43.55	44.ИПД
45.ИПА	46.ИПВ	47.ВП	48.1	49.+
50.1	51.0	52.0	53.±	54.+
55.С/П	56.Сх	57.БП	58.55	59.ИПД
60.ИПД	61.1	62.0	63.±	64.1
65.,	66.8	67.Ф1/х	68.-	69.1
70.ВП	71.7	72.+	73.ФВх	74.-
75.ПД	76.1	77.0	78.х	79.-
80.В/0				

Для демонстрации игр 1-3 в ячейки адресуемой памяти ПМК заносятся числа в соответствии с табл. 1, а для игр 4 и 5 - табл. 2.

Перед началом новой партии нажать В/0, затем С/П; ПМК сделает первый ход.

Ответный ход вводится в ПМК, после чего следует нажать С/П.

Завершая партию, ПМК сообщает о своей победе тремя выигрышными номерами или знаком "-" - о ничьей.



90

5

```

10 DIM X(10), A(5)
20 X(0)=
30 X(1)=
40 X(2)=
50 X(3)=
60 X(4)=
70 X(5)=
80 X(6)=
90 X(7)=
100 X(8)=
110 X(9)=
120 B=X(0): GOSUB 290
130 C=X(Y): D=C
140 IF C=0 THEN 150 ELSE 160
150 INPUT "ОБЩЕКА. ВАШ ХОД:";Y:GOTO 130
160 IF C<0 THEN D=-C
170 FOR I=1 TO 5
180 P=INT(D/10):A(I)=D-P*10:D=P: NEXT I
190 B=A(1):GOSUB 290
200 IF A(2)=Y THEN 230
210 PRINT "ВЫИГРЫВАЮ ПО ЛИНИИ ";
      ((X(0)*10+A(1))*10+A(2))

```

```

220 GOTO 310
230 B=A(3): GOSUB 290
240 IF A(4)=Y THEN 270
250 PRINT "ВЫИГРЫВАЮ ПО ЛИНИИ ";
      ((X(0)*10+A(3))*10+A(4))
260 GOTO 310
270 IF C<0 THEN PRINT "НИЧЬЯ. ХОЖУ";A(5)
      ELSE PRINT "ВЫИГРЫВАЮ ПО ЛИНИИ ";
      ((A(1)*10+A(3))*10+A(5))
280 GOTO 310
290 PRINT "ХОЖУ ";B
300 INPUT "ВАШ ХОД:";Y:RETURN
310 END

```

Для демонстрации игр 1-3 элементом массива X следует присвоить значения в соответствии с табл. 1, для игр 4 и 5 - с табл. 2.

#### Литература

1. Гарднер М. Есть идея! М.: Мир, 1982.
2. Гарднер М. Математические новеллы. М.: Мир, 1974.

## А. ЗАЙЦЕВ

# Спутники, компьютеры, образование

Новые информационные технологии (НИТ) — это развитие и повсеместное освоение не только ЭВМ, но и многих других нововведений, в первую очередь в области связи. В конце 1989 г. силами групп энтузиастов должны были быть запущены на круговую полярную орбиту 900 км сразу 6 компьютеризированных спутников, среди задач которых — и учебные. Надеемся, что некоторые возможности, предоставляемые ими, будут использованы и в СССР.

1. Спутник DOVE — digital orbiting voice encoder. Он содержит микро-ЭВМ и речевой процессор. Последний предполагается использовать как наглядное демонстрационное пособие при изучении основ космической связи. Для этого с командной наземной станции в него будут загружать короткие сообщения, составленные школьниками. Речевой процессор DOVE воспроизводит сообщения на португальском, английском и русском языках. Частота передатчика DOVE 145,875 МГц находится в пределах радиоловительского двухметрового УКВ-диапазона. Кроме речевых сообщений DOVE будет передавать обычную телеметрию о работе ЭВМ и других бортовых систем. Проект DOVE финансируется Бразильским союзом радиоловителей при поддержке радиоловителей из США [1]. Подробнее о нем будет рассказано ниже.

2. Спутник LUSAT. Его разработка и изготовление финансируются аргентинскими радиоловителями. Спутник предназначен для

проведения пакетной связи. Микро-ЭВМ имеет память порядка 4М байт, что позволяет записывать и передавать несколько сотен сообщений длиной до тысячи слов. Регистрирующая система имеет 4 канала в 2-метровом любительском диапазоне, передающая работает на одной частоте в диапазоне 430 МГц. Наземная станция представляет из себя УКВ-приемопередатчик 430/144 МГц, контроллер пакетной связи по протоколу AX-25 и микро-ЭВМ. Скорость передачи информации около 1200 бод. Детальная информация по пакетной связи содержится в [2, 3]. У нас в стране имеется более тысячи школьных коллективных радиостанций, поэтому реализация работы в режиме «электронной почты» между советскими школами или со школьниками в других странах представляется вполне реальной.

3. Спутник PACSAT. Этот спутник в основном подобен спутнику LUSAT. Его изготовление и запуск организованы американским Союзом радиоловителей.

4. Спутник WEBERSAT. Он разработан и изготовлен силами студентов Вебер-колледжа (г. Огден, штат Юта), имеет простую телекамеру, дающую цветное изображение Земли с разрешением 780×490 пикселей. С учетом оптики камеры и орбиты спутника кадр будет охватывать площадь 350×350 км. Изображение будет обрабатываться на борту микро-ЭВМ и передаваться на Землю в сжатом виде. Кроме телекамеры

на спутнике будет спектрометр, работающий в нескольких линиях видимой части спектра. Прием и передача данных будут вестись в 23-сантиметровом любительском диапазоне (1260—1270 МГц). Учебная программа спутника WEBERSAT рассчитана на студентов аэрокосмических специальностей, изучающих вопросы передачи и обработки информации по спутниковым линиям связи.

5. Спутники UOSAT D и E. Они разработаны и подготовлены группой студентов Университета Суррей (Англия), содержат несколько научных приборов: магнитометр, спектрометры, систему точного определения координат и т. д. Учебное назначение — проведение занятий со студентами аэрокосмических специальностей вузов и специализированными кружками радиолюбителей. Эта программа ведется уже несколько лет.

Из изложенного видно, что запуск «любительских» спутников открывает совершенно новые возможности для учебного процесса, базирующегося на информатике. Учитывая накопленный опыт по работе со спутниками, запущенные ранее, можно констатировать, что они позволяют решить несколько учебных задач:

разработать учебные программы по спутниковой связи для студентов и школьников, включая вопросы телеметрии, разработку бортовой аппаратуры, наземных комплексов приема и обработки данных и т. п.;

разработать простые школьные программы для развития интереса детей посредством прямого участия в приеме и передаче сообщений через спутники на основе простейших любительских радиостанций коллективного пользования;

расширить участие общественных организаций в образовательных программах за счет их привлечения к реализации спутниковых программ;

разработать учебные программы для школьников по мониторингу окружающей среды и экологической обстановки с помощью приборов, установленных на спутниках (магнитометр, камера съемки облачности, спектрофотометр и т. п.).

После решения технических вопросов организации УКВ-связи в пакетном режиме возникает задача формирования учебных заданий для школьников. Такие задания, на наш взгляд, должны включать следующие вопросы:

изучение основ спутниковой связи, включая схему построения связи, расчет элементов орбиты спутника, объем и скорость передаваемой информации и т. п.;

изучение использования спутников: для навигации, для народного хозяйства, для мониторинга окружающей среды;

изучение вопросов географии и геофизики

с помощью приборов на спутниках серии UOSAT и профессиональных (метеорологических «Метеор» и природопользовательских типа «Ландсат», «Природа», «Спот»);

обмен языковыми и учебными программами в виде писем посредством электронной почты со школьниками других школ и других стран;

участие в обеспечении связи в международных программах, общественных движениях, географических экспедициях в Арктике и Антарктике и т. п.

Все задания включают в себя несколько стадий: знакомство с литературой и обмен сведениями в классе, планирование, проведение реальной работы со спутниками, обработка данных и составление отчета о проделанной работе, обмен сведениями о проводимых экспериментах со школьниками других школ и других стран, сравнение уровня и результатов экспериментов, постановка и решение задач самостоятельно, проведение конкурса на постановку интересной задачи и оптимальное решение ее.

Очевидно, что энтузиасты реализации электронной почты для школьных коллективов должны оформиться в виде группы, обменивающейся информацией по этому кругу вопросов и оказывающей содействие в решении технических и организационных проблем. В качестве первого шага предлагаем всем, заинтересовавшимся пакетной связью и электронной почтой, писать нам по адресу: 142092, Московская обл., г. Троицк, Учебный центр «Байтик».

Не забудьте вложить почтовый конверт с обратным адресом для ответа!

Проект DOVE. Его образовательная направленность даст материал для изучения физики, астрономии, электроники, географии и иностранных языков. Спутник можно использовать также на многих общеобразовательных уроках — от начальной до высшей школы. В Новой Зеландии воспитатели детских садов, теперь обучающиеся для получения радиолюбительских разрешений, даже планируют использование DOVE для программ обучения чтению. Возможности специального голосового синтезатора DOVE позволяют школам с радиолюбительскими интересами использовать его как носитель сообщений в мировом масштабе.

Компьютер DOVE следит за состоянием спутника и сообщает свою температуру, электрические условия, на солнце он или в тени, какова скорость вращения. Эта информация, называемая телеметрией, может оказаться очень полезной для понимания школьниками условий, в которых работает космический объект. Телеметрия будет выдаваться на английском языке и чередоваться с другими сообщениями. Рабочая частота

DOVE — 145,875 МГц, мощность — 4 ватта. Сигнал можно принимать с помощью простого приемника со штыревой антенной. Большинство школ будут иметь возможность принимать чистый сигнал непосредственно в классных комнатах. Спутник движется по геосинхронной орбите, и слушать его можно с 10.30 до 22.30 по местному времени.

После запуска DOVE начнется специальный радиолокационный учебный проект, который продлится около года. Короткие сообщения, выражающие призывы школьников в защиту мира, будут переданы в эфир.

Директора школ, желающие принять участие в этой программе, должны сообщить своим учителям о проекте и организовать комитет по подготовке сообщений. Классные занятия будут включать подготовку сообщений для DOVE с последующим обсуждением значения исследований, возможностей укрепления мира как во всем мире, так и у себя дома. Попутно изучается мировая география и отмечаются те области мира, где в данный момент имеются конфликты (или случались ранее). Специально рассматривается, как конфликты разрушают целые семьи и уничтожают детей.

**Требования к сообщениям.** 1. Сообщения могут быть придуманы учениками в возрасте от 8 до 18 лет.

2. Сообщения должны быть длиной не более 40 слов. Нет ограничения на минимальную длину сообщения.

3. Ученики могут написать столько сообщений, сколько пожелают.

4. Так как передачи DOVE происходят на частотах радиолюбительских диапазонов, должно быть учтено пожелание государственных учреждений, что передачи не должны быть использованы в пропагандистских целях. Будьте осторожны при составлении текста сообщений. Вот несколько советов.

а) Избегайте сообщений, которые могут быть неоднозначно поняты, например: «Сражайтесь за мир!»

б) Сообщения должны быть глобальными по смыслу и не ориентироваться на локальные или региональные мировые спорные вопросы или конфликты.

в) Специфические особенности стран не должны упоминаться.

г) Хорошее сообщение должно отражать молодежный интерес или глобальную перспективу использования спутников.

д) Так как много мировых конфликтов в прошлом или настоящем базируются на религиозной почве, сообщения, в которых упоминается религия, должны составляться в общих выражениях.

е) Слова «ядерный», «третья мировая вой-

на», «звездные войны» должны быть исключены, так как для многих стран эти термины имеют специфическое значение.

Школьный комитет должен отобрать 10 сообщений, удовлетворяющих данным критериям. Имена и возраст учеников должны быть включены в сообщение. Следует отпечатать сообщения, а также их перевод на английский язык.

**Запись на магнитную ленту.** Запишите все ваши сообщения на новую стандартную кассету. Используйте наилучшую записывающую аппаратуру, доступную в вашей школе, и записывайте голоса школьников с расстояния около 25 см. Вы должны произвести чистую запись, удалив из нее затем шест бумаж или фоновые шумы. Посоветуйте школьникам говорить легко и точно, без излишней драматизации. Около 15 с тишины должны разделять сообщения. Начинать каждое сообщение со слов (на своем родном языке):

Меня зовут..., мне... лет, я живу в городе..., в Советском Союзе.

Сделайте на кассете пометку с адресом и датой и отправьте ее вместе с напечатанным текстом, переводами и сопроводительным письмом в центр сбора сообщений. Тщательно упакуйте кассету во избежание повреждений в пути. Отправляйте по адресу:

PROJECT DOVE  
c/o Richard C. Engin  
AMSAT Science Education Advisor  
421 N. Military  
Dearborn, MI 48124 U.S.A.

Персонал центра оставляет за собой право просмотра сообщений и последующего отбора. Однако тщательная подготовка сообщений и хорошая запись сделают это излишним. Подтверждение в получении будет направлено каждой школе, принявшей участие в проекте.

Записи отобранных сообщений будут переведены в цифровую форму, переданы на спутник и занесены в память компьютера. Затем голоса школьников будут передаваться в эфир. Каждый ученик, написавший сообщение, которое затем будет распространяться посредством DOVE, получит специальный сертификат от Бразильского общества радиолюбителей и информацию о том, как он может стать радиолюбителем. Нет никаких ограничений по срокам присылки сообщений.

К сожалению, не будет возможности уведомлять школы, когда их сообщения будут переданы посредством DOVE. Те, кто будет слушать передачи DOVE достаточно долго, получат полное представление о взглядах школьников на проблемы мира, и, когда они

услышат свои сообщения, последние будут выглядеть как часть общего движения, а не как предложение отдельной школы. Это подчеркнет и глобальную сущность радиоловительского увлечения. Сообщение не на английском языке будут сопровождаться переводом на английский.

Сообщения, приведенные здесь, были написаны в Crestwood High School Creative Writing Class, Dearborn Heights, Michigan, U. S. A.

The stars shine above us as we gaze with awe.  
We all have one hope and prayer... world  
peace.

*Jill Totsky, Age 17*

As I circle the world I see all nations. May

each get to know the other and grow together  
in peace and harmony.

Sample written by instructor

No more war.  
No more hate.  
Peace and love.  
Wouldn't life be great.

*Kelly Marphy, Age 16*

## Литература

1. *Лабугин Е.* Радиоловительские сети пакетной связи // Радио. 1988. № 12. С. 9—11.

2. *Лабугин Е.* Пакетная связь: протокол AX.25 // Радио. 1989. № 3. С. 10—13.

3. *Ensign E.* DOVE: A. Microsat classroom messenger // Proposal for AMSAT. October 1988.

С. ЛАНДО, О. МАНАКОВА

## Компьютер в летнем лагере

Авторы этой статьи в течение трех лет принимали участие в подготовке и проведении Международной детской компьютерной школы-лагеря (МДКШЛ) в Переславле-Залесском. В июне — июле 1989 г. они побывали в компьютерном лагере «Опыт Новой Англии» в США в качестве руководителей группы советских школьников. Группа принимала участие в обмене между двумя компьютерными лагерями, организованном Эриком Фенстером. Естественно, они не могли удержаться от сравнений, результат которых и попытались здесь отобразить.

Отметим прежде всего некоторые общие для обоих лагерей характерные черты: оба лагеря существуют недавно, в течение четырех лет; количество детей в одну смену в том и другом лагере от 120 до 160 человек; нормальной считается нагрузка 2—3 ребенка на преподавателя.

Американские коллеги неоднократно предупреждали нас о том, что по этому лагерю нельзя судить обо всех американских детских лагерях. То же самое можно сказать и про МДКШЛ в Переславле-Залесском.

### 1. Цели и задачи лагерей

Директор Айзек Баумфельд так формулирует основную цель своего лагеря: «Предоставить детям возможность научиться чему-нибудь стоящему за лето». Эта цель, по-видимому пользуется популярностью у родителей — лагерь планируется расширять.

Цели советского лагеря более разнообразны. Вот некоторые из них: разработка и

апробация различных методов применения компьютеров в образовании; обучение школьников, проявивших свои способности в какой-либо области научной деятельности; попытка построения модели будущего информационного общества, насыщенного компьютерами, средствами связи и передачи информации любому своему члену; общение детей разных стран и их совместная целенаправленная деятельность.

Отметим, что эти разнообразные цели нередко вступают между собой в противоречие и не всегда достижимы одновременно.

Поскольку цели двух лагерей различны, туда попадают дети с разными установками. Американский лагерь более развлекательный, детям предоставляется огромный выбор возможностей, которых они не имеют в школе (например, цирковая программа). Лагерь в Переславле-Залесском больше ориентирован на учебу, на получение научного результата. Он тоже предоставляет своим участникам большие возможности — пообщаться с известными учеными, талантливыми педагогами, получить доступ к хорошей современной технике.

### 2. Организатор лагеря. Международные связи

Американский лагерь представляет собой частное предприятие. Он существует на денежные взносы родителей. Плата составляет от 1025 долларов за ребенка за одну двухнедельную смену до 3700 долларов за 4 таких смены. Поэтому в лагерь попадают в основном дети из состоятельных семей. Среди них немало иностранцев, но детей из



Советского Союза лагерь принимал впервые. Впрочем, отдадим должное хозяевам, своей исключительности мы не ощущали, разве что о нас несколько больше заботились, стараясь показать страну.

МДКШЛ организуется Институтом программных систем АН СССР по инициативе его директора А. К. Айламазяна.

Пути набора детей в переславский лагерь так же многообразны, как и его цели. Мы берем школьников, показавших высокие результаты на различных конкурсах и предметных олимпиадах. Охотно принимаем хорошо себя проявивших участников предыдущих лагерей. В лагере обычно бывают группы младших школьников, необходимые для проведения педагогических экспериментов. Эти группы набираются, как правило, из переславских школьников.

Зарубежные участники лагеря обычно представлены группой школьников из г. Купертино (США) — побратима г. Переславля-Залесского, — и групп из Болгарии, Чехословакии, США, Польши, ФРГ, других стран, с которыми у Института программных систем АН СССР имеются научные связи. Отметим, что именно приглашение в лагерь группы купертинских школьников послужило толчком к установлению братских связей этого города с Переславлем.

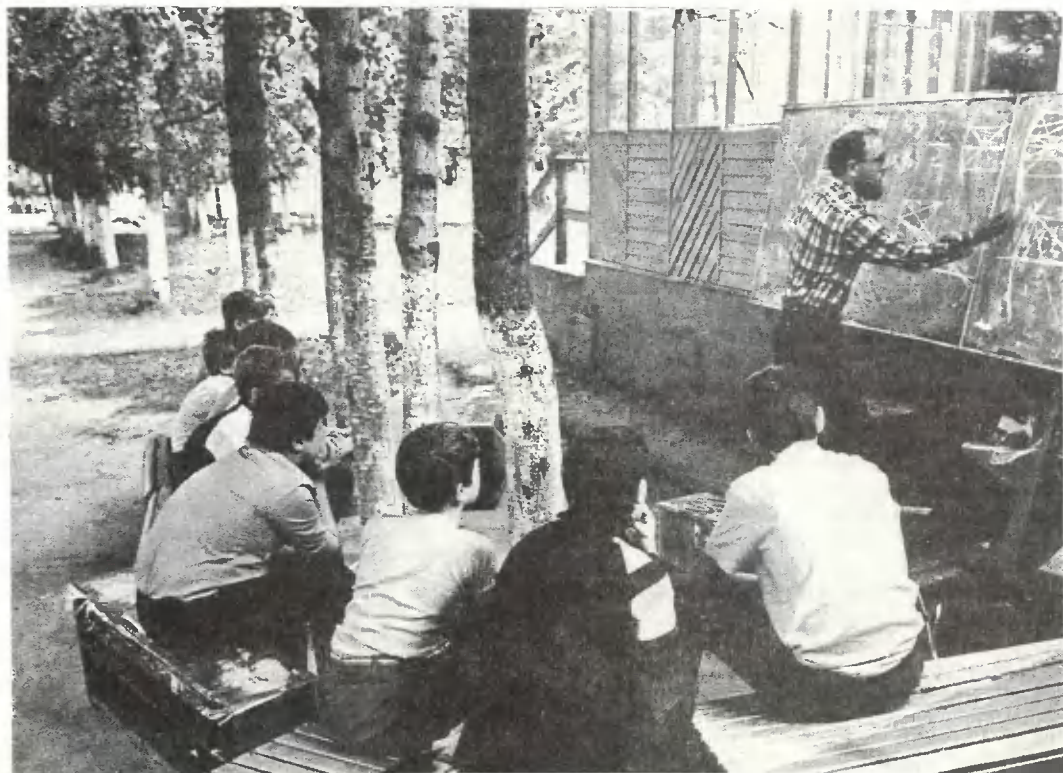
### 3. Организация занятий.

#### Подбор преподавателей

В каждом из двух лагерей школьник может выбрать себе вид деятельности, которым он хочет заниматься. В американском лагере набор этих видов чрезвычайно разнообразен. Отметим прежде всего его двойную специализацию: лагерь компьютерно-цирковой. Эти два столь различных направления неплохо уживаются друг с другом. Оба пользовались большой популярностью у советских школьников.

Помимо разнообразных элементов циркового искусства (жонглирование, акробатика, фокусы и проч.) лагерь предоставляет возможность для множества спортивных (большой теннис, плавание, гимнастика, водные лыжи) и культурных занятий (музыка, в том числе компьютерная, театр).

Содержание компьютерной программы, на наш взгляд, вполне традиционно. Лагерь неплохо оборудован персональными компьютерами: двенадцать машин типа IBM PC XT, около тридцати машин типа «Макинтош», несколько машин типа «Эппл II+». Двухнедельные курсы были в основном посвящены программированию и включали в себя программирование на Паскале, Бейсике, Си, Ассемблере для начинающих и продолжающих. Американские школьники мало инте-



ресовались компьютерами: у тех из них, кому компьютеры интересны, они есть дома, и у них нет потребности в дополнительном общении с ними. Поэтому основная часть компьютерной программы приходилась на советских ребят, многие из которых не были удовлетворены ее интенсивностью.

Особого внимания заслуживает организация занятий в лагере. Как нам кажется, этот опыт может быть применен очень широко. Приведем полностью план одного рабочего дня (в воскресенье занятий не было).

#### Распорядок дня

7.30 — подъем.

8.00 — завтрак.

8.30 — уборка помещений.

9.30—10.30 — 1-е главное занятие — обучение (занятия, работа в мастерских, обучение теннису, плаванию, спортивным играм, цирковому и театральному искусствам, ремеслам, музыке, танцам и т. д.).

96 10.45—11.45 — 1-е дополнительное занятие — выбирается ежедневно (активный отдых).

12.00 — ленч.

1.00—2.00 — 2-е главное занятие — обучение (занятия, работа в мастерских, обучение теннису, плаванию, спортивным играм, цирковому и театральному искусствам, ремеслам, музыке, танцам, и т. д.).

2.15—3.15 — 2-е дополнительное занятие — выбирается ежедневно (активный отдых).

3.30—4.30 — 3-е дополнительное занятие — выбирается ежедневно (активный отдых).

4.40—5.15 — тихий час, письма, звонки домой, тихие занятия.

5.15 — обед.

6.15—7.15 — 3-е главное занятие — обучение (занятия, работа в мастерских, обучение теннису, плаванию, спортивным играм, цирковому и театральному искусствам, ремеслам, музыке, танцам и т. д.)

7.15—7.45 — подготовка к вечернему мероприятию.

7.45 — вечернее мероприятие.

8.45 — открывается буфет; отдых, общение.

9.30—10.30 — отбой (в зависимости от возраста).

У каждого школьника ежедневно было шесть часов занятий. Три часа были главными, три — дополнительными. Главные занятия дети выбирали себе в первый день на всю смену. Так, ребенок мог выбрать себе Паскаль-1 в качестве первого занятия, теннис — как второе и фокусы — в качестве третьего и проходил, соответственно, двухнедельный курс Паскаля для начинающих, учился теннису и получал подготовку в области магии.

Дополнительные занятия выбирались ежедневно независимо от всего остального. Так, в один день можно было выбрать

два часа плавания и один час игры в теннис, а в другой день заниматься по часу музыкой, фокусами и компьютерными играми.

Соответственно, каждый преподаватель знал, что ежедневно на каждое главное занятие к нему приходит постоянная группа детей, а состав группы на каждое дополнительное занятие он узнавал утром. Ежедневная нагрузка на преподавателя составляла пять часов. Помимо этого большинство из них выполняли функции воспитателя в группе из 10—12 детей.

Приведенная нами схема оказалась достаточно гибкой. Например, руководительница театральной программы рекомендовала своей группе детей записываться к ней на первое главное и ежедневно на первое дополнительное занятия. Начиная на 15 мин раньше, она имела в результате 2,5 ч занятий ежедневно. Это позволило группе за две недели подготовить музыкальную постановку, которая с большим успехом была продемонстрирована всему лагерю в конце смены.

Набор преподавателей в американский лагерь ведется в основном по объявлениям, публикуемым в печати. Приглашаются, конечно, и участники проведения предыдущих лагерей.

Результат такого набора хорош в тех случаях, когда удается набрать профессионалов в той или иной области, умеющих работать с детьми. Так получилось, например, с цирковой, театральной, компьютерной программами.

В переславской школе за время ее проведения были опробованы различные организационные структуры. Из них устоялись две: кафедральная и система проектов.

Кафедральная структура в общем соответствует университетской. В первые два-три дня пребывания в лагере школьники с помощью преподавателей осуществляют выбор наиболее подходящей для себя кафедры, на которой и занимаются всю смену. Так как набор большей части детей производят преподаватели кафедр, то каждая из кафедр оказывается обеспеченной детьми в достаточной степени. Направления деятельности кафедр могут соответствовать привычному научному делению (таковы, например, кафедры физики, экспериментальной математики, экологии, лингвистики), или существенно отличаться от него (кафедра «Роботландия», кафедра «ProLogo», художественная мастерская).

Общим в деятельности всех кафедр является стремление найти направления нестандартного и эффективного использования компьютеров в обучении. При подобном подходе даже традиционные виды обучения

приобретают непривычный характер: например, математика становится экспериментальной наукой, что и отражено в названии кафедры.

Соответственно такому подходу организаторы лагеря стремятся приглашать преподавателей, занимающихся активной научной деятельностью в своей области, любящих работать с детьми, умеющих общаться с компьютером. Многие преподаватели лагеря являются сотрудниками ИПС АН СССР.

Метод проектов приобрел широкое распространение в американском школьном образовании. В переславском лагере он эффективно используется в сочетании с кафедральной структурой. Проект представляет собой законченную работу, выполняемую одним или группой школьников в составе до 10 детей. В проектах участвуют дети разных возрастов, начиная с 9—10 лет. Большинство проектов в лагере связаны с компьютером. Темы их могут быть весьма разнообразны: мультфильмы, написанные с помощью графического редактора, древняя русская архитектура, моделирование полета бумеранга или исследование фрактальных множеств.

Наиболее удачные из этих проектов при соответствующей методической разработке могут, на наш взгляд, быть использованы в школьном обучении, испытывающем не-

достаток в хорошо поставленных задачах к курсу информатики.

#### 4. Заключение

Организуемые по инициативе академика Е. П. Велихова обмены между компьютерными лагерями набирают силу. Их перспективность определяется наличием реальной базы для совместной деятельности детей, закладывающей основы естественного продолжения сотрудничества в различных областях. Подобные обмены представляются нам неотъемлемым атрибутом информационного века.

Наш визит в американский лагерь, а также участие иностранных школьников в МДКШЛ очень полезны и интересны как детям, так и преподавателям. Возможность получить информацию из первых рук, увидеть, чем интересуются, чем увлекаются (и чего не любят) твои сверстники и коллеги в другой стране, воспитанные в другой культуре,— вот что, на наш взгляд, самое ценное в таких обменах. Вместе с тем уже видны пути повышения эффективности совместной работы, сближения и более полного удовлетворения взаимных интересов. Мы надеемся, что приобретенный нами опыт сослужит добрую службу организаторам различных лагерей у нас в стране.

97

### Вниманию читателей!

Запросы относительно информации, опубликованной в рекламном объявлении кооператива «ПЕРСПЕКТИВА»

(«Информатика и образование», № 4, 1989), следует направлять по адресу:

129041, Москва, Б. Переяславская ул., д. 1, средняя школа № 1140.

Хозрасчетное подразделение кооператива «Перспектива», школьный кооператив «Интеграл-1140».

Телефон: 280-66-74.



Марийский государственный университет  
Марийское областное правление СНИО  
Госкомитет СССР по народному образованию  
в марте — апреле 1990 г.  
проводят школу-семинар

«Экспертные системы и Пролог в учебном процессе II»

Адрес для справок: 424001. Йошкар-Ола, площадь Ленина, д. 1,  
Марийский государственный университет. Телефон: 6-21-77.

ГРИГОРЬЕВ Сергей Георгиевич.

Главное учебно-методическое управление высшего образования,  
 Главное управление научно-исследовательских работ Гособразования СССР  
 совместно с фирмами АСКИ и «Синдзидайся Ко., Лтд» (Япония),  
 Коммерческим центром Гособразования СССР,  
 Научно-практическим центром «Образование»  
 при Московском государственном педагогическом институте им. В. И. Ленина

### объявляют открытый Всесоюзный конкурс на лучшую программу

для ПЭВМ типа КУВТ MSX-1, MSX-2 («Ямаха»), IBM PC  
 и совместимых с ними.

Участвовать в нем могут отдельные граждане и организации.  
*Программы на конкурс принимаются с 1 февраля по 31 марта 1990 г.,  
 подведение итогов — в конце июня 1990 г.*

Победителей ждут многочисленные призы и премии, в том числе  
 четыре первых премии (3000 долларов или профессиональная ПЭВМ),  
 поездки в Японию, поощрительные премии на сумму 10 000 рублей и т. д.

Базовые организации:

- конкурса программ для ПЭВМ «Ямаха» — МГПИ им. В. И. Ленина  
 (119435, Москва, М. Пироговская, 1; тел. 264-25-56, 264-45-74);
- конкурса программ для ПЭВМ типа IBM PC — НИИ ВШ АПН СССР  
 (103062, Москва, Подсосенский пер., 20; тел. 297-65-15, 297-55-26);
- конкурса программ для научных работ на ПЭВМ типа IBM PC —  
 Коммерческий центр Гособразования СССР  
 (129843, Москва, ул. Щелкина, 22; тел. 284-34-14, 288-46-70).

По этим же адресам можно получить всю информацию о конкурсе.  
 Ответы на вопросы к фирмам АСКИ и «Синдзидайся» можно получить  
 в Москве по тел. 230-08-83 («Синдзидайся»),

133-20-50 (УПЦ ВТ Октябрьского р-на),

в Новосибирске по тел. 45-09-37 (Центр «Информатика и дети»).

С победителями конкурса возможно заключение коммерческих  
 лицензионных контрактов на продажу программ в СССР и за рубежом.

В заявке на участие следует указать, что участник конкурса  
 является автором программы и не использовал в ней  
 программные продукты третьих лиц.

Предлагаемые на конкурс программы,

готовые к использованию в организации досуга, учебной работе  
 и научной деятельности, представляются в следующем виде:

программа на флоппи-диске с наклейкой, на которой указано  
 ее название, имя и адрес автора, тип ПЭВМ и требования к ней;  
 сопроводительная документация, включающая руководство с указанием  
 функций программы, правилами управления и другой  
 полезной информацией, заявку на участие в конкурсе с подписью,  
 подтверждающей согласие с перечисленными условиями.

## МОЛОДЕЖНАЯ ИНИЦИАТИВА

### В НОМЕРЕ

- АЛМА-АТИНСКИЕ ШКОЛЬНИКИ ПРОГРАММИРУЮТ КОМПЬЮТЕРНЫЕ ИГРЫ НА ВОРОНЕЖСКОМ ТЕЛЕВИДЕНИИ
- МЕЖШКОЛЬНОЕ КОНСТРУКТОРСКОЕ БЮРО В МОСКВЕ
- ГЛАЗАМИ СОЦИОЛОГА

### Кто поставит задачу?

Алма-атинским мальчишкам, и девочкам даже в самый разгар жаркого лета есть чем заняться. И не только потому, что можно ловить бабочек или пускать кораблики в говорливых арыках. Просто им повезло, что в их прекрасном городе есть дом, где их всегда ждут,— Республиканский дворец пионеров. А то, что одна из его комнат оборудована компьютерами «Ямаха», приводит сюда еще несколько сотен ребят. Да, именно с такой цифры в списках желающих научиться работать на «электронном чуде XX века» начинается занятия кружок программирования каждой осенью. А самые увлеченные не расстаются с любимым делом даже летом.

«Способные ребята,— говорит руководитель кружка программирования алма-атинского республиканского дворца пионеров Светлана Леонидовна Кирпанева,— быстро осваивают все доступ-

ные возможности компьютера и ждут серьезного задания».

В первое время, когда школьники начинают изучать азы программирования, руководители кружка могут им дать и дают много нового и полезного. Но уже через несколько месяцев, о чем сами кружководы заявляют с нескрываемой гордостью, ребята ведут разговор с компьютером и со своими руководителями на равных. Правда, вслед за этим возникают проблемы. Не всегда находится сильная, интересная и практически полезная задача. Ведь для того, чтобы разработанная школьниками программа нашла широкое применение и за пределами компьютерного класса, в котором была разработана, она должна быть не только аккуратно и правильно написана, но и выполнять какую-то полезную работу.

Но пока алма-атинским

юным программистам краснеть за свои работы не приходится. Всюду, куда бы они ни посылали свои программы, к ним проявляется повышенный интерес. Это и авиатренажер «Экипаж» Жени Ковешникова и Сергея Чуканова из СШ № 54, и серия лингвистических программ, посвященных изучению русского языка и тренировке мыслить логически («Сконструируй слово»), и многие другие.

Как правило, программы хорошо прокомментированы, в них рационально используются все возможности компьютера: цвет, звук, графика, полиэкранные возможности. Правда, вспоминая недавний международный конкурс по математике и программированию, проходивший в Алма-Ате, педагоги отмечают существенную разницу в стиле программирования наших и болгарских ребят, для которых важно не только написание работоспо-

собной программы, выполняющей поставленное задание, но и качество исходного алгоритма, структурность программирования, продуманная организация структур данных.

Поэтому в программах болгарских ребят было легко разобраться, оценить их красоту и, конечно, правильность. А затраты на постановку задачи обернулись для них большой экономией времени на исправление ошибок и отладку.

Написание больших программ, разрабатываемых коллективами программистов, немыслимо без определенной культуры программирования. Поэтому коллективное написание одной большой задачи могло бы стать хорошей школой, приучающей ребят к коллективному мышлению и выработке совместной технологии программирования. Заодно это позволило бы взяться не за игровую или обучающую программу скромных размеров, а произвести на свет серьезный и полезный программный комплекс. Кстати, во Дворце пионеров недавно организована Малая академия наук школьников Казахстана, которую курирует академик В. М. Амербаев. Можно надеяться, что профессиональные программисты вычислительного центра Академии наук Казахской ССР помогут ребятам в выборе проблемы.

А пока задачи берутся из самых невероятных мест: из детских и юношеских журналов, с «маминой работы», из головы. Был случай, когда сюжет программы подказала книжка детских стихов по правилам дорожного движения. Воодушевленные прочитанным, старшеклассники Алексей Мозаренко и Андрей Толоченко написали программу, помогающую малышам осваивать азбуку дорожного движения.

А ученик XI класса РФМШ Антон Ершов подумал о своих друзьях, мечтающих научиться играть на клавишных музыкальных инструментах, и написал программу «Музыкальный редактор», которая не только демонстрирует акустические возможности «Ямахи», но и позволяет обучиться нотной грамоте. Севший за пульт компьютера ученик, нажимая на клавиши и подбирая на слух мелодию, может увидеть ее записанной нотами на экране дисплея. Если мелодия понравилась, ее можно запомнить, а позже исправить или дополнить новыми электронными звуками. Пока редактор позволяет писать только одноголосую музыку, но Антон работает сейчас над новой, многоголосой версией «Музыкального редактора».

Помимо развлекательных или учебных в кружке разрабатываются и «деловые» программы. Их назначение — облегчить работу с компьютером деловым людям. Информационно-поисковая система Димы Беклемишева (X класс РФМШ) позволяет организовать данные почти как в настоящей базе данных, обеспечивает их хранение, сортировку, поиск, изменение. Подобные программы входят в стандартный комплект программного обеспечения профессиональных персональных компьютеров наряду с текстовыми и графическими редакторами, документаторами, электронными таблицами, трансляторами. Информационно-поисковая система — лишь первое звено в цепочке утилит, делающих компьютер доступным каждому. И быть может, стоит подумать о том, чтобы открыть в журнале рубрику «Учимся писать систему», в которой Дима рассказал бы о том, как писать простые базы данных, а Антон — как

устроен его «Музыкальный редактор». Тогда каждый смог бы оборудовать свой компьютер целым комплексом учебных утилит. И поверьте, работать с изготовленным собственными руками транслятором со своего, пусть очень простого, языка не менее интересно, чем играть с самой интересной программой.

А для того чтобы эти программы получились полноценными, разрабатывать их надо на современной компьютерной технике. Именно поэтому пионеры в первую очередь необходимо обеспечивать лучшей вычислительной техникой. Ведь не секрет, что в школах компьютеры не всегда используются «на все сто». Кроме того, к ним, как правило, закрыт широкий доступ всех желающих научиться программированию. А отсутствие методических пособий по проведению занятий в группах информатики или недостаток квалифицированных преподавателей приводит порой к тому, что компьютерный класс превращается в компьютерную игротеху. А во Дворцах пионеров все заинтересованные ребята могут получить консультацию, обменяться идеями. Кроме того, упростится организация различных конкурсов программистов, одной из трудностей которых является невозможность проверки работоспособности программ из-за отсутствия компьютеров нужного типа.

Необходимо изменить и практику проведения конкурсов. Ведь все люди, использующие компьютеры в своей деятельности, делятся на «пользователей» и «программистов». Поэтому и обучение информатике, и соревнования в мастерстве общения с компьютером надо проводить раздельно для программистов и пользова-

телей, которые с большим мастерством, чем остальные работают в редакторе или с документатором, базой данных или электронной таблицей. Существуют же талантливые конструкторы автомобилей и высококлассные автогонщики. А есть и самородки, не только в кустарных условиях собирающие свои автомобили, но и умеющие на них ездить. Хотя никто больше этими автомобилями не пользуется.

А с идеями ведь не везде так плохо.

Быть может, стоит помимо конкурса программ объявить еще и конкурс идей, на основе которых можно написать полезные программы? Авторы лучших из них в идеале стоило бы награждать компьютерами для того, чтобы у них появилась возможность реализации своих замечательных идей.

А пока можно посоветовать всем, кто хочет найти новые идеи, если их не вдохновили задачи написания учебной операционной системы, продолжать регулярно просматривать все доступные научно-популярные журналы, такие, как «В мире науки» (рубрика «Занимательный компьютер»), «Квант», «Наука и жизнь», а также не пренебрегать другими не менее полезными изданиями, и прежде всего «Информатикой и образованием». А для начала попробуйте реализовать программу Дж. Конвея «Жизнь» или собрать панно из плиток Трюше (см. В мире науки. 1989. № 9. С. 81.). И конечно, пишите нам о своих самых невероятных идеях.

**В. ТИХОНОВ**

## Телекомпьютерные игры

Одним из нестандартных путей повышения компьютерной грамотности является цикл телевизионных передач «Компьютерные игры», подготовленный Воронежским телевидением и факультетом прикладной математики и механики Воронежского госуниверситета. По содержанию телекомпьютерные игры разработаны на основе опыта межвузовских компьютерных игр, которые регулярно в течение трех последних лет проводятся среди студентов г. Воронежа. Передачи рассчитаны на широкие круги зрителей: школьников, учащихся техникумов и ПТУ, студентов младших курсов вузов, рабочих, служащих, домохозяек.

Несмотря на то, что цикл предназначен в основном для непрофессиональных пользователей ЭВМ, многие программисты и электронщики, будущие специалисты по прикладной математике (по их собственным отзывам) нашли в компьютерных играх много полезного. Кроме того, квалифицированные пользователи ЭВМ пробуют свои силы в составлении вопросов и заданий для следующих передач.

В каждой компьютерной игре участвуют две команды: игроки и соперники. Игроки — это команда школы, вуза, техникума, ПТУ, любого предприятия, успешно выполнившая домашнее задание. Соперники — это составители вопросов и заданий, болельщики в зале и у экранов телевизоров, т. е. те, кто не входит в команду игроков. Состав команды игроков — 10—15 человек, состав команды соперников неограничен.

В первой телекомпьютерной игре в качестве игроков выступала команда средней

школы № 89 г. Воронежа: учащиеся IV—X классов, учительница начальных классов, старшая пионервожатая. Готовил команду к играм учитель информатики Ю. Сушков. О своем желании быть игроками следующих телекомпьютерных игр заявили команды многих школ г. Воронежа, команда Воронежского высшего военного училища радиоэлектроники, команда строительного института, техникума железнодорожного транспорта, сборная команда г. Борисоглебска Воронежской области, различные клубы и многие другие.

В первой телекомпьютерной игре, которая длилась 45 мин, было разыграно 15 вопросов и заданий. За правильные ответы победители награждены микрокалькуляторами различных марок, электронными часами, транзисторами, которые предоставили спонсоры игр — предприятия г. Воронежа. Если игроки не могли дать верный ответ, то приз получал автор вопроса, а баллы — команда соперников.

Часть вопросов и заданий была ориентирована на использование ЭВМ, некоторые рассчитаны на общую компьютерную эрудицию. Игры проходили во Дворце пионеров и школьников, где были установлены БК-0010, «Агат», «Электроника-85», «Ямаха», микрокалькуляторы. Вопросы и задания, разнообразные по форме, содержанию и сложности, требовали от 40 с до 12 мин для ответа и оценивались различными количеством баллов (от 0,5 до 3 баллов).

Судила встречу игроков и соперников экспертная комиссия, состоявшая из профессиональных пользовате-

лей ЭВМ. Диспетчером состязаний был компьютер «Агат», для которого разработана специальная программа «Бейся», позволяющая вести игру, следить за ее ходом и фиксировать протокол соревнований. Так, «Бейся» выбирает задание для очередного тура, следит за временем его выполнения, ведет счет игры.

Одна из целей, которую преследовали телекомпьютерные игры, состояла в том, чтобы болельщики и телезрители были не пассивными наблюдателями, а активными участниками. Поэтому в решении некоторых вопросов принимают участие не только игроки, но и болельщики в зале и даже телезрители. Одно из заданий игр состояло в разработке программы, которая по известной, введенной в память БК-0010 итоговой таблице игр чемпионата страны второй лиги по футболу определяла, какая из команд перейдет в первую лигу. Игроки это задание выполняли на Бейсике. Болельщики и телезрители составляли программу на алгоритмическом языке. Болельщики в зале должны были сдать программу через 20 мин, игроки — через 12 мин, а телезрители через — 1 ч после окончания передачи. Честно говоря, мы не ожидали особой активности телезрителей и были приятно удивлены большим количеством программ телезрителей. Пусть не все программы оказались верными, но зато все, кто пришел после окончания телепередачи, живо обсуждали возможные варианты решения не только на алгоритмическом языке, но и на Бейсике, Фортране, Паскале, получили возможность отладить программы на персональных ЭВМ. Это вселяет в нас уверенность, что после телепередачи ряды друзей

информатики существенно пополнятся. Отчасти такую активность телезрителей можно отнести и на счет Оганеза Мхитаряна — кумира воронежских футбольных болельщиков, который сформулировал на телеигре это задание (воронежская команда «Факел» переходит в следующем году в первую лигу).

Для игроков и телезрителей предназначен вопрос на общую компьютерную эрудицию. Мы назвали этот вопрос музыкальным, так как он задается после показа видеоклипа, озвученного современной электронной музыкой. Формулировка музыкального вопроса: «Способность синтезаторов копировать звучание инструментов и даже человеческого голоса сделала их необходимым атрибутом звукозаписывающих студий. Если со звуком что-то не в порядке, то его можно добавить, убавить или, вообще, слепить совершенно новый. Ведь любой звук — это крупница информации, а заряженный информацией компьютер способен сделать что угодно. Компьютеризации пока еще не поддается один-единственный из рюковых инструментов. Через 1 мин вы должны назвать этот инструмент, который не поддается компьютеризации». Телезрители в течение этой минуты могли позвонить по указанным номерам и сообщить свои ответы. Интересно, что от нескольких телезрителей поступили правильные ответы (это акустическая гитара), тогда как игрокам музыкальный вопрос оказался не по силам.

Одно из заданий телекомпьютерных игр состояло в проверке знаний игроков по синтаксису языка Бейсик, которая осуществлялась в режиме диалога игрок—ЭВМ «Агат». Для этой цели было разработано соответ-

ствующее программное обеспечение, согласно которому из банка вопросов автоматически выбирались несколько вопросов (результативного, выборочного или конструируемого типов) и выводились на экран дисплея. Игроки вводили свои ответы с терминала. Их ответы по программно контролировались. В конце диалога, который длился 2 мин, ЭВМ оценивала знания игроков. Каждое действие игроков сопровождалось необходимыми текстовыми сообщениями, выводимыми на дисплей.

Когда диспетчер — «Агат» выбрал задание № 7, зазвучала музыка мазурки, и танцоры (победители зонального конкурса бальных танцев) исполнили фрагмент этого старинного танца, который помимо красоты, развлекательности имел и смысловую нагрузку. После исполнения танца партнер снял фрак и положил его на стол, где находились компьютер, жилет и панталоны. Далее последовал вопрос: «Что общего у всех этих предметов?», на обдумывание которого игроки получили 1 мин. Очень интересно шло обсуждение, было несколько версий, но ни одной верной. Хотя достаточно было вспомнить строки А. С. Пушкина из «Евгения Онегина»:

Фрак, панталоны и жилет —  
Всех этих слов по-русски нет.

Как известно, компьютер тоже не русское слово.

ПЭВМ «Электроника-85» использована в телепередаче как средство для разгадывания кроссворда, включающего распространенные термины из области информатики на английском языке. Разработана соответствующая программная поддержка, которая организует выдачу на экран дисплея любого вопроса кроссворда, номер которого задает программа «Бейся» «Агат» —



диспетчера; проверку правильности внесения символов очередного слова в определенные позиции кроссворда; ограничивает время работы игроков с одним вопросом (40 с). По правилам игры игроки один раз могут воспользоваться словарем английских терминов. В случае, если игроки затрудняются отгадать какое-либо слово, вопрос адресуется болельщикам в зале. Программное обеспечение разработано на языке «Модула» и включает модуль, облегчающий составление любого другого кроссворда пользователю, лишь в общих чертах знакомому с программной поддержкой «кроссворд».

Составляя набор заданий к игре (а все вопросы имеют автора, фамилия которого называется вместе с вопросом), мы стремились сделать ее интересной для многих телезрителей, разнообразив форму передачи материала. Поэтому наряду с музыкальными, танцевальными, игровыми, кроссвордными вопросами, о которых речь шла выше, у нас было и немое кино. Именно таким образом разыграна сцена, поддерживающая вопрос об изобретении Германа Холлерита.

Микрокалькуляторы предназначались для определения спелости арбуза. В распоряжение игрокам на 2 мин предоставлялись весы, сантиметровая лента, арбуз, микрокалькуляторы и формула зависимости веса спелого арбуза ( $m$ ) от длины ( $l$ ) его обхвата ( $m = 17 \cdot 10^{-6} \cdot l^3$ ).

Для того, чтобы показать, что вычислительной техникой надо пользоваться разумно и существуют правила быстрого устного счета, болельщикам было предложено поспорить в умножении, используя калькулятор, и объяснить, почему умножить

любое четырехзначное число на 9999 (пятизначное на 99999) можно устно.

Оживление игроков вызвал вынос подноса с чашками киселя и блюдцем с хреном. После чего прозвучал вопрос «Какое отношение имеет содержание чашек и блюдца к знаменитым школьным учебникам по математике, по которым учились ваши родители?». Такой, казалось бы, легкий вопрос (это учебник А. П. Киселева, который родился,

жил и работал в селе Хреновое Ново-Усманского района Воронежской области) вызвал бурное обсуждение, рассматривались несколько версий, но ни одна из них не была доведена до верного ответа.

Первая телепередача из серии компьютерных игр понравилась и игрокам, и болельщикам в зале, и, что особенно важно, телезрителям, для которых она и предначалась.

О. УСКОВА

## Школьники становятся программистами

Межшкольное конструкторское бюро (МКБ) организовано в 1985 г. при Институте проблем информатики АН СССР. Цель — создание профессионального программного продукта силами учащихся школ. В конструкторском бюро работают ученики VII—X классов из различных районов Москвы и пригородов. Сегодня МКБ объединяет 65 членов, включая наших выпускников — ныне студентов ведущих вузов.

**Наши ресурсы.** МКБ располагается в отдельной комнате площадью 25 м<sup>2</sup>, имеет пять персональных компьютеров «Ямаха», оснащенных дисковыми и цветными дисплеями, и один принтер. Дело даже не в том оборудовании, которым мы располагаем, а в благожелательном отношении дирекции института и администрации РУНО Октябрьского района (в здании которого мы размещаемся). Добрые отношения позволяют МКБ работать 363 дня в году с 9 до 21 часа. Это приходится подчеркивать потому, что ни одно молодежное образование из известных нам не может позволить себе такой

режим из-за различного рода надуманных ограничений.

Сегодня МКБ тиражирует разработанную школьниками первую версию системы проектирования мультфильмов, машинных игр и учебных пособий. Выпущена книга «Система проектирования мультфильмов, машинных игр и учебных пособий», авторами которой являются школьники.

Одновременно с разработкой члены МКБ участвуют в течение года в обслуживании различных мероприятий МК ВЛКСМ, Госкомитета по информатике и вычислительной технике, Госкомитета по народному образованию, Министерства народного образования РСФСР, АН СССР, Октябрьского райкома партии. В летнее время члены МКБ обслуживают два пионерских лагеря.

Нередко члены МКБ привлекаются к участию в различных международных мероприятиях. В 1989 г. Филипп Бахрамов был в командировке в Кенфордской школе в Англии, Андрей Сарычев выезжал в Болгарию для участия в Международной школе по информатике, Александр Грун-

цев приглашен на первый всемирный молодежный фестиваль по информатике.

Кирилл Чурюканов, Дима Черемушкин, Максим Буланов, Сергей Бакулин, Борис Рогачев обслуживали мероприятия в пионерском лагере «Орленок», наладили выпуск стенной компьютерной газеты и обеспечили голосование с обработкой результатов на ЭВМ с возможностью поименного голосования. Все необходимые программы были написаны непосредственно в лагере.

Каждый учащийся VII—IX классов имеет право проработать в МКБ в течение двух месяцев испытательного срока. По истечении срока решается вопрос о зачислении.

Во время испытательного срока будущий коллега должен доказать свои способности к самостоятельной работе. Он имеет доступ к литературе и вычислительной машине. Дается 5—6 заданий, сложность которых зависит от уровня поступающего. За время испытательного срока любая помощь запрещена. Выполненные задания строго проверяются на правильность и стиль программирования. За это время выясняются черты характера и степень серьезности увлечения программированием. Если претендент выдержал испытательный срок, он становится членом МКБ.

#### **Обязанности члена МКБ.**

Член МКБ обязан:

учиться на «хорошо» и «отлично»;

в IX—X классах поступить в вечернюю школу при избранном вузе;

научиться благожелательно относиться к товарищам по работе в МКБ;

доводить до логического конца порученное дело;

знать, что результатом работы является программа с документацией, оформленной в соответствии со стан-

дартами и пригодная для тиражирования.

Недопустимо для члена МКБ вранье и зазнайство. Член МКБ, нарушивший эти правила, отчисляется.

Участие в различных мероприятиях — путь к самостоятельности.

Как уже говорилось, члены МКБ участвуют в городских, всесоюзных и международных мероприятиях. До тех пор пока не сложилось дружное квалифицированное ядро МКБ, автору этих строк приходилось сопровождать выезжающую группу. Когда сложилось ядро, все поездки стали делом самих ребят.

Если учесть, что школьники выезжают с вычислительными машинами, за которые несут ответственность, как и за всю работу, то читатель поймет, насколько трудно было пойти на такой шаг.

Сегодня можно сказать, что такая постановка вопроса полностью себя оправдала.

Оценка уровня знаний в МКБ несколько отличается от общепринятой.

Первый уровень предполагает знание одного или нескольких языков программирования и умение их использовать для решения простых задач.

Второй уровень предполагает умение создать программу достаточной сложности, представляющую интерес для широкого круга потребителей, и выпустить

документацию по программе.

Третий уровень предполагает умение самому найти достойную задачу и довести решение до логического конца.

Первого уровня знаний достигают все члены МКБ.

Второго уровня достигают 50—60 % занимающихся у нас ребят.

За четыре года существования МКБ третьего уровня знаний достигли шесть человек. Определенного уровня знаний член МКБ достигает самостоятельно, используя консультационную помощь своих товарищей.

Наши молодые коллеги, достигшие второго уровня, становятся способными работать совместно по единому сложному проекту. (Таким образом была создана система проектирования мультфильмов, машинных игр и учебных пособий, и сегодня более совершенные и сложные программные комплексы.)

Автор будет благодарен читателю, если тот за перечислением фактов и несколько радужным тоном статьи увидит ежедневную напряженную работу молодого увлеченного коллектива, у которого есть свои просчеты и ошибки и свои радости. Радости прежде всего в творчестве, что и отличает от порой унылых занятий.

Телефон для справок в Москве: 132-19-74.

**Д. НЕПОЧАТЫХ**

## **Компьютер и отдых детей**

Одной из интересных форм приобщения детей к миру компьютеров являются летние детские компьютерные лагеря. Их количество ежегодно растет, а география расширяется. Специфика и преимущества таких компьютерных лагерей, на наш

взгляд, заключаются в следующем. Во-первых, здесь ребята полностью погружены в «компьютерную среду». Это особенно существенно в условиях дефицита вычислительной техники. Во-вторых, большое значение имеет привлечение высоко-

квалифицированных педагогов, различных специалистов. При стабильном руководстве лагеря, свободном подборе кадров возможно формирование очень сильного коллектива, который будет в состоянии реализовать различные педагогические и научные инновации. В-третьих, социально-психологическая атмосфера лагеря, где на основе общего увлечения компьютерами неформально общаются с преподавателями и между собой дети разного уровня подготовки, возраста, культуры, национальностей, — все это значительно облегчает и усиливает процесс обучения и воспитания.

Одним из первых в стране четыре года назад был создан международный детский компьютерный лагерь в г. Переславле-Залесском. Постоянным куратором и активным помощником лагеря в решении многочисленных проблем является Институт программных систем АН СССР. Лагерь относительно небольшой (110 детей), хорошо оснащен компьютерной техникой. Для изучения социальных аспектов влияния информатизации на детей летом 1989 г. был проведен анкетный опрос. В выборочной совокупности оказались школьники восьми национальностей (в частности, поляки, болгары, финны) в возрасте от 12 до 16 лет.

В целом участники лагеря оказались хорошо знакомыми с ЭВМ: 55,4 % опрошенных умеют работать на ЭВМ, а 44,6 % могут самостоятельно написать программу для решения интересующих их задач. Основные формы приобщения ребят к компьютерам: школьные занятия по информатике (у 67,9 % опрошенных ЭВМ имеются в школах), посещение компьютерного клуба (кружка) — почти каж-

дый третий (28,6 %) регулярно посещает их и влияние родителей (у каждого десятого — 10,7 % — дома имеются личные компьютеры, а почти у половины — 44,6 % — родители пользуются ЭВМ).

Для большинства школьников начинающаяся информатизация представляется пока не очень важным направлением (50,0 % считают, что есть более важные, например проблемы экономики). Чего же ожидают опрошенные от предстоящей информатизации общества? В первую очередь улучшения условий труда (73,2 %) и повышения качества обучения и образования (66,1 %). Каждый третий надеется, что информатизация позволит улучшить социально-бытовые условия жизни людей (39,3 %).

Больше всего опасаются школьники распространения стрессов и «компьютерных болезней» (39,3 %), усиления эгоизма, индивидуализма, отчуждения друг от друга (23,2 %), общей деформации духовных ценностей в негативную сторону (17,9 %). В целом же эти мнения отражают широко распространенные в массовой печати представления о последствиях информатизации.

Очень интересны представления школьников об основных требованиях к человеку, предъявляемых информатизацией. По их мнению, главное — уровень знаний и образованности человека (так считают 32,1 % опрошенных), затем — умение работать на ЭВМ и с информацией (16,1 %), внимание (12,5 %), крепкие нервы и выдержка (8,9 %). Видно, что по рангу предпочтений у школьников умение работать с ЭВМ на втором месте, а на первом — интеллектуальные способности человека. Опрошенные ребята в

большинстве своем хорошо владеют компьютером, очевидно, поэтому для них не представляется важным такое свойство, как умение работать с ЭВМ. Несмотря на это, многие (71,4 %) хотят и дальше учиться программированию, а также совершенствовать свои навыки работы на ЭВМ (76,8 %).

В лагере дети имеют возможность много общаться с компьютером. Каждый четвертый (23,2 %) в среднем работает на ЭВМ более 3 часов ежедневно, каждый восьмой (12,5 %) — от 2 до 3 часов каждый день. Тем не менее по существующим нормам и правилам каждый день оператор ЭВМ может работать за дисплеем не более 4 часов. Такое интенсивное общение с компьютером вызывает сильную усталость лишь у 1,8 % опрошенных, у 14,3 % — небольшую усталость, у 35,7 % — незначительный дискомфорт, а 5,4 % чувствуют себя даже бодрее и лучше. В основном страдают глаза — у 8,9 % сильно, у 28,6 % — устают, но не очень сильно.

Уникальные возможности компьютера и интенсивное общение с ним вырабатывают у ребят соответствующий «образ» ЭВМ. Самое главное достоинство компьютера — это то, что с ним интересно играть (80,4 % опрошенных так считают). Но при этом достаточно критически оцениваются возможности ЭВМ — 82,1 % считают, что компьютеры не всегда правы; 53,6 % — что компьютеры не всегда делают то, что им прикажут. Компьютеры у опрошенных школьников не вызывают чувство неудобства или стеснения — 83,9 % считают, что ЭВМ может пользоваться любой желающий.

При оценке предпочтительности марок компьютеров только 17,9 % хотели приобрести советскую ЭВМ;

в полтора раза больше (25,0 %), хотели бы иметь персональный компьютер социалистических стран и в четыре раза больше (62,5 %) — ЭВМ капиталистических стран.

Основные мотивы приобретения компьютера — для самообразования (39,3 %), помощи в учебе (37,5 %) и для игр (35,7 %). Личный компьютер влияет на структуру интересов и потребностей: по оценкам владельцем ЭВМ, каждый восьмой (12,5 %) из них стал меньше смотреть телепередач, каждый одиннадцатый (8,9 %) — меньше ходить в кино. В то же время 23,2 % ребят свободно владеют английским и еще 32,1 % —

владеют этим языком со словарем (на нем даются описания инструкций, программ, правил и т. д.). Для сравнения, только 1,8 % свободно владеют французским и немецким языками, т. е. распространение новых информационных технологий сопровождается и распространением языка страны — лидера в разработке ЭВМ и программного обеспечения.

Как уже отмечалось, многие опрошенные (44,6 %) пишут самостоятельно программы. Каждый пятый из этого числа (9,0 %) пытался внедрить свои разработки в практику, но удавалось это сделать только одному из трех (3,6 %). В то же

время в большинстве своем ребята активные, готовые отстаивать свои идеи и разработки — только 16,1 % из них отказались бы от борьбы за внедрение своего новшества, а 44,6 % не уступили бы.

В плане общежитейских дел ребята больше всего довольны взаимоотношениями в семье (51,8 %) и с друзьями (сокласниками) — 50,0 %; меньше всего довольны уровнем своей правовой защищенности (50,0 %), условиями для отдыха (30,4 %), отношениями с учителями и содержанием учебы в школе (23,2 %).

**В. ЧЕРЕДНИЧЕНКО,**  
канд. эконом. наук,  
НИЦ ВКШ при ЦК ВЛКСМ

## Программная система «ПЕРСОПЛАН»

«Персоплан» — компьютерная система поддержки принятия решения.

«Персоплан» можно использовать в разнообразных ситуациях выбора при наличии нескольких противоречивых критериев качества решения.

В методической схеме «Персоплана» реализован оригинальный алгоритм структуризации компонент решения: пользователь сам может настроить «Персоплан» на интересующую его проблемную область, вводя в компьютер в ходе специального диалога формулировки мотивационных и инструментальных конструктов — критериев качества решения. Теоретическая модель, лежащая в основе «Персоплана», соединяет в себе достижения психологической теории решений (максимизация функций ожидаемой полезности) и психологии личностных конструктов (техника репертуарных решеток).

«Персоплан» позволяет ввести в компьютер и оценить до 120 наименований целей и мотивов выбора. Система снабжена тремя файлами со специализированными контрольными списками и матрицами связей, делающими «Персоплан» готовым к применению (в качестве наполненной экспертной системы) для трех типовых проблемных ситуаций: выбор профессии (профорентация), выбор супруга (брачное консультирование), выбор товара (маркетинг). Таким образом, «Персоплан» может быть эффективно использован психологами и социологами в широком классе практических ситуаций оказания клиенту помощи в принятии решения. Кроме профессиональных применений «Персопланом» может воспользоваться любой пользователь персонального компьютера, готовый к развитию культуры самоанализа.

Меню-ориентированный диалог, многочисленные подсказки и «хелп-режим» (выдача на экран глав руководства по запросу в ходе работы с системой) облегчает освоение и обеспечивает комфорт в работе с «Персопланом». Аппаратные средства — IBM PC-AT/XT и совместимые модели. Емкость комплекса — 150К байт. Имеется гарантия. Ориентировочная цена — 300 руб. по б/р.

**Заявки (гарантийные письма) направлять по адресу:**

107066. Москва, ул. Спартаковская, 13,  
генеральному директору СП «Диалог» П. С. Зрелову.  
Тел.: 939-38-90, 203-66-28.

Джейн НОКС

Национальная Академия наук США

## Что могут дать компьютеры педагогике: Взгляд из американской школы

Сегодня в США компьютеры используют не только программисты-профессионалы, но и секретари, учителя, студенты, родители вместе со своими маленькими детьми. Как пишет один специалист по вопросам дошкольного обучения: «Предсказывают, что в самом ближайшем будущем в каждой второй американской семье будет домашний персональный компьютер, которым смогут пользоваться и маленькие дети» [1]. При школах, как правило, имеются специальные вечерние компьютерные курсы, которые позволяют всем желающим познакомиться с элементами новой информационной технологии: использованием текстового редактора, базами данных, средствами для развития своих детей, способами телекоммуникации и т. п.

Компьютер как инструмент общения открывает новый этап развития современного человека. По мнению некоторых американских ученых, компьютер может не только изменять образ и стиль мышления человека, но и создать предпосылки для формирования нового подхода к образованию. Главная цель настоящей статьи — привести аргументы в пользу этого мнения.

В 1984 г. на внедрение ЭВМ в образование США было израсходовано 60 млн. долларов. [2]. Основное внимание сейчас обращается на введение ЭВМ в учебный процесс в качестве его существенной составной части. Как использование ЭВМ в процессе обучения меняет традиционную ситуацию в классе и роль учителя, как подготовить учителя к этой новой для него роли — вот вопросы, которые стоят сегодня на повестке дня.

Впервые компьютеры появились в школах в 60-х гг. Это были главным образом боль-

шие мощные компьютеры. Затем появились персональные компьютеры и начали разрабатываться обучающие программы. Однако, согласно данным, полученным 5 лет назад, только 5 % этих программ оказались эффективными [3]. Сегодня положение меняется: психологи и педагоги начинают сотрудничать с программистами. Качество программ становится все лучше и все больше отвечает особенностям познавательных процессов у детей на разных этапах их развития.

В 80-х гг. основное внимание обращается на подготовку учителей к использованию ЭВМ в учебном процессе. Сегодня к учителям обращен призыв: «Начните действовать. Попытайтесь понять, что происхо-

107





108 дит. Примите участие в использовании этого нового технического средства. Начните работать в педагогике, ориентированной на применение компьютера, с тем, чтобы контроль над ситуацией и ответственность за нее остались бы в ваших руках. Опытный, образованный в этой сфере учитель и гибкие технологии открывают прекрасную возможность — создать для последующих поколений условия формирования совершенно нового опыта» [4].

К настоящему времени еще не проведено системного исследования влияния применения ЭВМ на процесс образования, а имеющиеся результаты не позволяют построить целостную картину необходимого подхода к этой проблеме [5]. Методика использования ЭВМ при работе с детьми и сама система «ученик — компьютер — учитель» еще недостаточно разработаны.

Компьютеризация образования в США начиналась с изучения в старших классах программирования. В последние годы благодаря появлению на рынке большого числа уже готовых фирменных обучающих программ процесс компьютеризации распространился практически на все учебные предметы и охватывает все возрастные ступени, включая дошкольников. Имеются убедительные доказательства в пользу того, что компьютер обеспечивает возможности для развития наглядно-образного мышления, моторных и вербально-коммуникативных навыков, целеустремленности и социализации [6]. Последнее является особенно важным фактором в обучении маленьких детей, и компьютер может способствовать активизации вербального взаимодействия учащихся, особенно в тех ситуациях, когда дети работают парами или в группе.

Можно выделить два основных направления создания педагогических программных средств. Первое определяется тем, что детям предлагается выполнять на компьютере ряд упражнений из обычного школьного курса. Целью использования компьютера при этом является заучивание некото-

рых правил или автоматизация каких-то навыков. Как правило, это связано с механическим повторением учебного материала.

Этот подход развивается большей частью авторами, работающими для коммерческих компаний, которые выпускают готовые обучающие программы. В настоящее время в США имеется много программ такого типа для обучения, например, английскому языку, математике, биологии и т. д. Использование этих программ ориентировано на закрепление тех навыков, которые до некоторой степени уже сформированы на уроке в классе.

Стандартные коммерческие пакеты программ для изучения математики или английского языка довольно легко интегрировать в существующие школьные курсы.

Достоинства этого подхода в том, что учащийся работает на ЭВМ индивидуально, получает немедленный ответ на каждое действие. В обычной ситуации работы с классом, где находятся более 20 человек и один учитель такая обратная связь невозможна. Традиционный урок не позволяет обеспечить столь высокого уровня индивидуальной работы, скорее, имеет место монолог со стороны учителя. Компьютер, в отличие от других средств обучения детей (книга, магнитофонная запись, телевизор, и т. п.), ...обладает способностью реагировать. Характер его реакции определяется поведением того, кто с ним работает. Поскольку ответы ЭВМ определяются намерениями и желаниями того, кто с ним работает, то между компьютером и человеком может возникать обмен информацией» [7]. Более того, работая с компьютером, учащийся получает ощущение, что он сам управляет своей учебной деятельностью. Это очень важный психологический фактор, обеспечивающий более активный режим обучения.

Включение ЭВМ в процесс школьного обучения приводит к необходимости пересмотра вопроса о роли учителя. Традиционное представление о том, что учитель полностью управляет ребенком и говорит ему, что и как он должен делать, должно быть модифицировано. Сохранение старого подхода может свести на нет тот эффект, который связан с использованием компьютера. Согласно новому пониманию учитель выполняет роль режиссера, который только направляет ученика. Он создает условия и предпосылки для осуществления деятельности ученика, который эту деятельность актуализирует.

Второе направление разработки компьютерных программ для школьников связано с использованием ЭВМ в специально соз-

даваемой среде. В его основе лежит представление о том, что компьютер можно использовать для развития творческих способностей ребенка. Главным аспектом обучения здесь становится не заучивание правил, а сам процесс мышления. Наиболее значительный вклад в работы этого направления внесли С. Пейперт [10], С. Веир [4, 7] и Р. Тинкер (технические педагогические исследовательские центры — TERC). Под их руководством разработаны и широко используются программные среды: система LOGO и ее дальнейшее развитие LOGOWRITER; LEGO — LOGO; лаборатории, основанные на микрокомпьютерах, т. е. MBL, где дети сами становятся исследователями, открывающими законы математики, биологии, физики. Эти открытия тесно связаны с реальной окружающей их физической средой. Работая с LOGOWRITER, ребенок сам открывает мир геометрии и математики, сообразуясь сначала с движениями своего собственного тела, затем с реальной игрушкой — черепахой и наконец с двумерной черепахой на экране.

Работая в MBL и используя компьютер, соединенный с системой датчиков для измерения света, влажности, температуры и т. д., ребенок исследует разные актуальные научные проблемы: кислотные дожди, радоновый газ, погоду и т. п. Результаты своих исследований он передает школьникам, живущим в другом географическом районе, и обменивается с ними мнениями о значимости своих данных по компьютерной сети. Этот подход можно обозначить как исследовательский: ученик сам открывает что-то новое. Такое использование ЭВМ в обучении приводит к необходимости построения новой концепции образования: погружение в свой собственный опыт. «Быть учеником — значит исследовать окружающий мир, задавать собственные вопросы, пытаться выяснить, что произойдет, если...» [9].

Для учителя, желающего посмотреть на процесс обучения со стороны, компьютер как бы визуализирует умственную деятельность учащегося и дает возможность следить за ней. В ситуации традиционного обучения наблюдать за особенностями работы каждого из 25 или больше детей просто невозможно. Использование ЭВМ снимает эти проблемы. Учитель может следить за работой каждого, видеть, как ребенок понимает задачу, расценивает ее и т. п. Наконец, у учителя есть возможность, проанализировав накопленные в памяти компьютера данные о работе ребенка, получить представление о том, как он думает. Компьютер дает возможность намного более полного и глубокого, чем при традиционном обучении, понимания процесса умственного

развития ребенка. «Микрокомпьютеры позволяют вскрыть структуру умственной деятельности пользователей. Благодаря компьютеру мы получаем окно, позволяющее заглянуть в мыслительный процесс ученика, и это является наиболее важным преимуществом компьютеризованного обучения» [9].

Обычно выделяются два способа познания: вербальный, основанный на речи, и зрительный, основанный на изображении. У одних более развиты вербальные способности, у других — зрительно-образные. То огромное значение, которое придается языковым умениям в традиционном обучении, ставит в худшее положение детей с преимущественным развитием зрительно-образных способностей. Определенные возможности при обучении этой категории учащихся могут быть связаны с использованием компьютера, позволяющего сместить акцент в педагогической практике в сторону процессов, основанных на зрительной переработке информации. В последнее время ширится признание того, что пространственное мышление играет важную роль в овладении математикой, но до сих пор развитию навыков формального пространственного мышления уделяется мало внимания в учебном процессе.

Использование компьютера в педагогической практике может в значительной мере восполнить этот пробел. Для тех детей, которые отличаются более развитым зрительно-образным мышлением, компьютер — это как раз то, что им нужно.

Работа с графическим материалом побуждает таких детей к обсуждению того, что они видят. Это имеет особое значение при обучении детей с задержкой речевого развития или в тех случаях, когда речевые звуки оказываются просто недоступными для детей и представляют собой всего лишь сотрясение воздуха, т. е. с глухими, аутичными и некоторыми умственно отсталыми детьми.

Программа LOGOWRITER ориентирована на формирование представления о программировании в самом общем смысле и позволяет ребенку продумывать отдельные шаги и их последовательность, чтобы достичь своей цели. Таким образом, ребенок сам управляет процессом обучения: сам себе ставит задачу (как это должно быть в реальной жизни) и сам находит пути ее решения. Традиционная учебная ситуация переворачивается: не компьютер программирует работу ребенка, а ребенок работу компьютера. Он сам программирует свое поведение, моделируя его на компьютере. «А обучая компьютер как думать, ребенок

открывает, как он думает сам» [10].

Важным преимуществом этой программы является то, что ребенок решает задачи не на абстрактных схемах, а с помощью наглядно-чувственного объекта — черепашки, поведением которой он управляет. Обучение с помощью LOGOWRITER начинается в реальной, живой жизни и к ней возвращается. Ребенок должен соотносить движения собственного тела с движениями черепахи на экране, поставив себя на место черепашки. Для этого, прежде чем он начинает управлять движениями черепахи по экрану, он играет с реальной игрушечной черепахой, передвигая ее по полу, либо сам играет роль черепахи, передвигаясь по комнате, либо «превращает» в черепаху учителя, управляя его передвижением, с помощью устных команд — команд, в каком направлении двигаться, сколько шагов идти вперед или назад, куда повернуть.

110 Такое «перевоплощение» оказывается особенно эффективным при работе с аутичными и детьми с другими аномалиями развития. Это важно не только с точки зрения приобретения навыков «управления» объектом, который передвигается на экране компьютера в соответствии с их собственными движениями, но и имеет определенный корректирующий эффект, обеспечивая преодоление их общей моторной скованности, заторможенности, пассивности. В силу традиционных установок и представлений, да и особенностей воспитания в семье, доброта окружающих проявляется как немедленная готовность предоставить детям все самое необходимое. В результате развивается что-то вроде «воспитанной (приобретенной) пассивности». Работа с компьютером, связан-

ная с собственной двигательной активностью, с ощущением себя как действующего персонажа (черепашки), помогает преодолеть эту пассивность. Работая с программами типа LOGOWRITER, ребенок вместо привычного ожидания, чтобы для него все сделали другие, попадает в ситуацию, которой он может управлять сам. Это в известном смысле приводит к изменению образа Я: из постоянно ждущего помощи от других он превращается в ребенка, который самостоятельно ищет решения и самостоятельно их находит.

Любой ребенок, обучаясь, становится исследователем, открывателем. Это означает, что ребенок учится делать выводы и обобщать, исходя из собственного опыта манипуляции с конкретными предметами, окружающими его.

Важно отметить, что такого рода предметная деятельность, направленная на решение задач с помощью компьютера, облегчает процесс взаимодействия и способствует социализации при работе детей в парах или в группах. Дети предлагают различные решения какой-либо задачи, например: построить такую машину из пластмассовых кирпичей, которая при объединении с компьютером и выполнении совместно выбранных учениками команд LOGO будет передвигаться по полу с максимальной возможной скоростью. В ходе такой деятельности ребенок часто обнаруживает инициативу и качества лидера, которые не проявились у него в традиционной ситуации классной комнаты.

Рассмотрим три конкретных примера, где компьютер оказал реальную помощь трудному ученику в массовой школе США.

Первая история. 15-летний Джон С. из школы в Кларктауне (штат Нью-Йорк) был хронически неуспевающим (58 пропущенных дней в течение учебного года, неудовлетворительные оценки по всем предметам школьного курса за предыдущий учебный год). Когда он попробовал работать с программой LEGO — LOGO в MBL для изучения физики, у него внезапно проснулся интерес к наукам естественного цикла. Его первым заданием было сконструировать машину, способную как можно быстрее двигаться вверх по наклонной плоскости. Джон приступил к выполнению задания с удовольствием, поскольку воспринимал его как решение «реальной» проблемы. Когда задание было выполнено, оказалось, что он способен перенести приобретенные в процессе его выполнения знания в собственную жизненную ситуацию. Он теперь знал принцип работы переключателя скоростей на своем велосипеде, знал, что





необходимо сделать для увеличения скорости и мощности, и знал взаимосвязи между скоростью и усилием, когда приходится въезжать на крутой подъем. Эти знания Джон теперь может использовать и в других жизненных ситуациях, связанных с передвижением транспорта.

Вторая история. Разительная перемена в поведении 19-летнего Джерри, исключенного из одной школы и поступившего в другую «альтернативную» экспериментальную школу в Шарлотсвилле (штат Виргиния), произошла, когда он начал работать с программой LEGO — LOGO в рамках предметов естественного цикла. Джерри, до того славившийся своим невосдержанным, агрессивным характером, склонный к антисоциальным поступкам, внезапно оказался в положении лидера среди своих соучеников. Это произошло после того, как он смог объяснить им смысл устройства переключения скоростей и характер соотношения между скоростями. Дело в том, что он сам к этому моменту усвоил в MBL такие понятия, как трение, торможение, скорость и мощность в процессе конструирования машинки сначала с передним, а затем с задним приводом. Эти понятия стали вполне доступны ему, когда он начал работать, реализуя предметные виды деятельности в этой богатой компьютерной среде. Обращаясь к собственному конкретному опыту, Джерри блестяще объяснял и разрешал трудности, с которыми детям приходилось сталкиваться, и, естественно, был очень горд этим. Остальные учащиеся, работающие в этой MBL, в свою очередь, испытывали удовольствие от того, что с ними говорит их сверстник на их собственном языке.

Третья история. 9-летняя девочка с выраженной умственной отсталостью начала заниматься по специальной программе в условиях массовой школы в маленьком городке Джей (штат Мэн). Было известно, что у нее имеется грубое нарушение внимания и практически отсутствует речь. Однако, когда она приступила к работе с компьютером, который был установлен в классной комнате, ежедневно от 15 минут до 1 часа, в ее развитии обнаружился весьма значительный прогресс. Подвижные, большие картинки, появлявшиеся на экране ЭВМ, немедленно привлекали ее внимание. Мультисенсорный вывод информации также способствовал ускорению процессов обучения и постижению окружающего — компьютер позволял объединить динамическую графику, письменное слово и его звуковой аналог, воспроизводящийся с помощью синтезатора речи. Добавим к этому использование языка жестов, которые сопровождали каждое



слово и высказывание. Все это способствовало быстрому развитию речевых навыков у девочки. Она начала активно участвовать в диалоге ребенок — компьютер — учитель, в особенности когда ей удавалось вызвать на экран компьютера ее любимые объекты: мороженое, телевизор, желтый цвет, плачущего медведя. Изображения этих предметов, как правило, вызывали ее восторженную вербальную реакцию.

Анализ ситуации в американской школе показывает, что многие дети начинают отставать, потому что им приходится работать в рамках модели, ориентированной в системе противопоставления «правильно — неправильно». Основное значение использования сред типа LEGO — LOGO и MBL в педагогической практике связано с появлением новых подходов к образованию. Чрезвычайно важную роль в этом новом представлении обучения играет исправление ошибок в работе по созданию программ [11]. Часто, но не всегда для этого требуется анализ и углубленное понимание программы и ее частей. В ходе этого процесса очень часто получается что-то интересное и неожиданное, и ребенок может таким образом научиться чему-то новому.

Главный вопрос, который возникает при отладке программы, связан не с оценкой правильности или ошибочности, а с возможностью развертывания процесса отладки программы самим учеником. Высказать замечания о принципиальной важности самовлеченности в путь познания, а не в достижение цели нас побуждают два примера работы с девочками 9 лет, которые впервые сели за компьютер.

Первая — заикающаяся девочка должна была собрать на экране дисплея изображения, употребляя команды LOGOWRITER. Никакой подсказки, какую картинку собрать, она не получила. Наоборот, на вопрос, что ей хочется собрать, она ответила: «Дом с крышей». В первой попытке она строила дом, рисуя черепашкой прямоугольник и треугольник. Из-за ошибки она оставила «дырку» в прямоугольной стене дома, и, когда стала заполнять этот прямоугольник «кирпичами», «кирпичи» стали вываливаться из дырки и заполнили весь экран. Во второй попытке, не стирая дома и дорисовав замкнутые прямоугольные стены (без дырки), она успешно заполнила его розовым цветом, который ярко отличал дом от синего фона с кирпичами. Девочка очень гордилась своей работой.

Вторая девочка — слабослышащая — хотела нарисовать пейзаж с солнцем на небе. Цвет при изображении солнца она по ошибке выбрала сиреневый, хотя хотела желтый. Перерисовывать изображение она не захотела, но дала своей картине интерпретацию, для которой вполне подходит сиреневое солнце: она нарисовала домик и над ним написала: «Это солнце, солнце сиреневое, оно сказочное, это дом гномика».

Человеку важен не конец дороги, а путь туда, как написал Достоевский. Согласно этому, сам процесс доведения программы до конца, часто требующий глубокого анализа и понимания каждого шага по пути, доставляет нам большое удовольствие. Нет

нужды доказывать, что в отсутствии страха сделать то, что может оцениваться нашим учителем или сверстником как ошибка, мы все — и дети, и взрослые — обнаружили бы значительно больше инициативы, желания открывать новое и добиваться поставленных целей в самых различных сферах жизни.

Литература

1. Йорки Т. ЭВМ в детском саду // Перспективы: Вопросы образования. ЮНЕСКО. 1987. № 4, С. 96.
2. Там же.
3. Aiken R. Teachers and Computers. What is the key component // Paper Presented at AEC (Automatization of the Educational System) in Secondary and High Schools. Institute Kurchatova. M., 1989, May 26.
4. Weir S. Cultivating Minds: A LOGO Case Book. Cambridge, Mass. 1987. P. 7.
5. Ocko S., Papert S., B Resnick M. LEGO, Logo and Science // Technology and Learning. 1988. 2 (1), P. 1—3.
6. Йорк Т. С. 98—101
7. Weip S. The Computer in Schools: Machine as Humanizer // Symposium: Computers in Classroom Instruction. Harvard Educational Review. 1989. Vol. 59. Nov. 1. P. 61.
8. Barclay T. Coping with Inquiry. Hands On // Newsletter of the Technical Education Research Centers. 1987. 10 (1), P. 6.
9. Weir S. Cultivating Minds: A LOGO Case Book. Cambridge, Mass. 1987. P. 7.
10. Papert S. Mindstorms: Children, Computers and Powerful Ideas. N. Y., 1980. P. 19.
11. Papert S. Mindstorms: Children, Computers and Powerful Ideas. N. Y., 1980. P. 19.

Что дешевле?

Как известно, деловые люди умеют считать деньги. И если уж японские производители автомобилей решились установить компьютер в чрево своего детища, то только для того, чтобы он приносил выгоду, работая на пределе возможностей. А в современном автомобиле настоящих дел даже для ЭВМ немало. Одно из наиболее важных на сегодня — экономичное использование топлива при всех режимах работы двигателя.

Компьютер, специально созданный для этого, следит за составом



приготовленной в карбюраторе топливной смеси и, если он не соответствует нагрузке на двигатель, изменяет его. Необходимая для такой работы информация (скорость движения, нагрузка, положение рулевого колеса и др. — селовой заслонки, качество топлива) поступает во встроенный компьютер с соответствующих датчиков. А он на основе этих данных принимает решение о не-

обходимом составе топливно-воздушной смеси для каждого из четырех цилиндров двигателя. Кроме того, в ЭВМ заложены программы, позволяющие питать двигатель наиболее рациональным образом на дорогах различного качества и при различных режимах вождения.

Бортовой компьютер может решать такие относительно простые задачи, как слежение за исправностью тормозной системы и обеспечение оптимального режима торможения, управление световыми приборами автомобиля, диагностика и сигнализация о различных неисправностях.

## Сердитое письмо

Двойственное впечатление оставил заключительный тур II Всесоюзной олимпиады школьников по информатике. Теплая встреча гостей олимпиады, интересная культурная программа, прогулки по прекрасному весеннему Минску — и, увы, вызывающая недоумение работа жюри. Сомнительными представляются и подбор задач, и их формулировки, и система оценок работ участников. Но — по порядку.

На теоретическом туре вторая задача предлагалась в следующей формулировке:

«Структура простого предложения имеет вид: (определение 1) подлежащее сказуемое (определение 2) (дополнение) (обстоятельство).

Члены в 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, глаголы, прилагательные, наречия».

Попробуем понять, что требуется от участника олимпиады. Мы видим, что определена некоторая конструкция русского языка (простое предложение), части предложения которой могут выбираться из перечисленных в условии частей речи. Требуется сформулировать правила русского языка, по которым может осуществляться такой выбор, и построить реализующий его алгоритм для словаря, определенного участником. При этом неясно, формулировать ли правила для выбранного словаря или общие правила и в связи с этим — что делать с окончаниями. Неясно также, какие сочетания необязательных частей предложения считать допустимыми, а главное, как быть, если теоретических знаний по русскому языку (которые, как известно, слабо

связаны с уровнем фактической грамотности) недостаточно для решения предложенной задачи (например, как быть участникам из союзных республик). На разборе задач выяснилось, что предложенная выше интерпретация условия никак не связана с тем, что имело в виду жюри олимпиады. На самом деле была дана некоторая абстрактная конструкция, названная для простоты «простым предложением», названия частей речи и частей предложения тоже никак не были связаны с семантикой русского языка и поименованы так для «удобства», а что все это такое, должен был определить сам участник.

Я готов согласиться с утверждением жюри о том, что на практике постановка задачи представляет собой зачастую более сложную проблему, чем ее решение на ЭВМ, но можно ли из-за этого ставить участника олимпиады в положение программиста, которому заказчик не может сформулировать свои требования?

Странным выглядит подбор задач для практического тура.

Первая задача была неоднократно разобрана в литературе, в частности популярной (например, в книгах Н. Я. Виленкина «Комбинаторика» и «Популярная комбинаторика»).

Вторая же была очень трудной, относилась к комбинаторной теории групп, точнее — теории перестановок, и место ей на студенческой олимпиаде по математике. На разборе автор задачи слово «группа» не произнес, однако, хотя решение полностью изложено не было, объяснение начал с поиска инвариантов и выписывания орбит.

Неоправданным представляется и беспре-

центное решение жюри оценивать все задачи и теоретического, и практического туров одинаковым числом баллов. Если учесть еще дополнительные баллы, начислявшиеся за хорошие решения, то в лучшем положении оказались участники, решавшие наиболее простые задачи. Так, в первой задаче практического тура участники, верно применившие рекуррентную формулу, получили 10 баллов, а те, что обратили внимание на довольно быстро возникающие ошибки, округления и справились с этим, реализовав «длинную» арифметику, получили дополнительные баллы (до 5). Единственный же участник, верно решивший первую задачу практического тура и, понятно, больше ничего сделать не успевший, получил всего 10 баллов, после того как член жюри и автор этой задачи объявил на разборе о большой весовой функции за ее верное решение. Жюри обосновывает свою удивительную позицию тем, что не по его вине работы были дешифрованы до того, как приняли решение о весовой функции (а по положению о проведении олимпиады оценки нельзя менять после определения авторства работ иначе, как апелляции). В ответ же на предложения определить весовую функцию без учета авторства работ, как то обычно и делается (например, максимальная оценка, полученная за задачу, делится на среднюю по участникам олимпиады оценку для получения весового коэффи-

циента по данной задаче; есть и другие известные методы), жюри заявило, что вопрос был поставлен на голосование и решение принято отрицательное. Интересно при этом следующее: если по положению оценку менять нельзя, то как же можно было ставить вопрос на голосование? А если можно, почему было принято такое странное решение? Потому, что так жюри проще?

Необходимо отметить и несовпадение требований к работам, сформулированных в памятках участникам, с теми, которыми руководствовались жюри при оценке работ. Так, от участников на практическом туре требовали подробных комментариев в исходных текстах программ, на что тратилось драгоценное время, а при проверке оценивались только результаты работы представленных программ и исходные тексты не рассматривались.

Хотя список претензий к работе жюри этим не исчерпывается, сказанного достаточно для того, чтобы при проведении III Всесоюзной олимпиады по информатике вопрос о составе оргкомитета и жюри олимпиады был рассмотрен более тщательно и гласно и опасные прецеденты не превратились в плохую традицию.

**Е. М. КУЗНИЦКИЙ,**  
руководитель Московской команды  
на II Всесоюзной олимпиаде  
по информатике

## Личность учителя

*Учитель не тот, кто учит,  
а тот, у кого учатся.*  
А. Куманев

Представьте себе, что завтра вы приходите в школу, в которуб введен новый принцип: отныне учащиеся могут выбирать не только предметы, но и преподавателя, у которого хотели бы учиться. Признайтесь, так ли вы уверены во всех своих учениках, в их выборе? Ведь не секрет, что предпочтение того или иного предмета (даже если ребенок посещает его по обязанности) в значительной степени связано со значимостью для ученика личности учителя.

Если учитель нравится, то и предмет изучается легко, с большим коэффициентом полезного действия. Не последнюю роль играет и личность учителя при выборе будущей профессии. Имея природные данные в какой-либо предметной области, но неглубокого, безразличного учителя, ученик нахо-

дит наставника или на стороне, или приходит в эту область много позже, преодолев в своем сознании связанные с личностью учителя неприятные ассоциации.

В процессе преподавания обязательно происходит воздействие индивидуальности педагога на ученика. Он сообщает не просто значения понятий, но и свое личное понимание и отношение к предмету. Без этой важнейшей стороны нет воспитания.

Теперь вообразите, что в школу приходит неодушевленный «учитель» — компьютер. Он может излагать фактический материал не хуже самого эрудированного предметника, оценить знания точно и беспристрастно, проследить ход решения задачи и выдать подсказки, провести любой тип урока, заметить все применяемые ныне ТСО.

Но. С компьютером нельзя установить мостик духовной близости, сопереживания. Он не дает нравственной оценки поведения ученика как личности, не проводит

адекватного психофизиологического анализа состояния ученика в момент обучения, не делает настройку своих педагогических средств воздействия для мотивации конкретного учебного занятия, не контролирует динамику изменения значимости для ученика получаемой информации. Компьютер не может расставлять акценты в системе ценностей человека, обогащать его социальным опытом, развивать эмоционально и духовно.

И по мере того как в образовании будут приходить наши технические помощники — компьютеры и педагогические компьютерные программы, все больше ребенок будет нуждаться и ценить личное общение с педагогом. А когда компьютер станет привычен, как мел и классная доска, учителей серых, ограниченных, работающих нетворчески, просто не должно остаться. Их «выживет» компьютер, соревнования с ним они не выдержат.

Как же, спросите вы, нужно работать, чтобы избежать предсказанного фиаско? Ответ покажется очень простым — без насилия над собственной личностью, без насилия над личностью воспитанника, с уважением его права быть самим собой.

Мне приходилось наблюдать, как порой «портится» молодой учитель и как к худшему меняется отношение к нему учеников. Поначалу, не владея еще в полной мере приемами педагогического мастерства, молодой учитель тем не менее покоряет ребят своей открытостью, добротой, отсутствием чопорной дистанции и незащищенностью. У него нет еще того «забрала», за которым истинного человеческого лица не разглядеть. В то же время наши застойные «традиции», не терпящие большого отклонения от среднего, от общепринятого,

стараясь выравнять, усреднить всех и каждого. В школе они имеют большую инерцию.

Через некоторое время молодой учитель, уже как все, по праву сильного применяет психологическое давление, «держит дисциплину», знает и умеет применять систему мер воздействия и наказаний. Он опустил «забрало», он доволен собой, так как начал приживаться в коллективе учителей, находящихся по другую от детей, их проблем и переживаний сторону баррикады.

Да простят мне мои коллеги эти зарисовки с натуры. Не во всех педагогических коллективах принято носить «железные доспехи». Если у вас в школе в норме искренность, взаимное уважение и всепобеждающая доброта во взаимоотношениях с учениками, вам повезло. В другой обстановке молодым хотелось бы пожелать не спешить «усреднять» себя. Нужно возвращать свою личность в тесном контакте с учениками и коллегами, взаимно обогащаясь в процессе общения.

И компьютер, с его техническими возможностями: скоростью поиска и обработки информации, цветографическим отображением, музыкальным звучанием — станет вашим другом и помощником. А компьютерные программы, вобравшие в себя всю мудрость известных систем и методов обучения, фактический материал, содержания основной и углубленной программ, построенных в соответствии с самыми прогрессивными организационными формами, помогут работать творчески, оставят больше времени и сил для решения гуманитарных педагогических задач.

**Н. ПАХОМОВА,**  
учитель СШ № 444  
Москва

115

## Доступно уже сегодня

Можно ли спорить с теми, кто говорит, что школа должна получать все лучшее, в том числе и вычислительную технику? Нельзя. А можно ли надеяться, что школы будут получать если не IBM PC, то хотя бы КУВТы с дисководом на каждом рабочем месте? Увы, нет. Поэтому, мечтая о светлом будущем, нужно не забывать о реальностях сегодняшнего дня.

Наша группа из сектора аппаратных разработок Центра информатики и вычислительной техники Казанского государственного университета в составе С. Ситникова, О. Киселева и автора этого письма разработала компьютерный комплекс с управляемыми кассетными накопителями на базе

магнитофона «Маяк-233», впрочем, может использоваться и любой другой бытовой кассетный магнитофон с электромагнитным управлением и датчиком числа оборотов подкассетника. Управляется магнитофон специальным драйвером через контроллер, выполненный в виде платы, вставляемой в разъем управляющего порта «УП» БК-0010.

Удобства работы на таком комплексе немногим уступают удобству работы с дисководами. К тому же у него есть одна интересная возможность, которая должна заинтересовать педагогов. БК-0010 может отыскать на МЛ не только компьютерный файл, но и место, где заранее записана некоторая звуковая информация для чело-

века (музыка, текст), и включить магнитофон на воспроизведение. Возможно, эта методическая новинка окажется полезной на уроках.

Комплект программного обеспечения комплекса включает кроме управляющей программы магнитофона сокращенный Бейсик объемом 8К байт, занимающий адресное пространство Фокала. Управляющая программа заносится в дополнительное внешнее ОЗУ, занимающее адресное пространство ПЗУ пользователя объемом 8К байт. Эта дополнительная внешняя память на 16К байт выполнена в виде отдельного блока, который вставляется в разъем общей шины на задней стенке БК-0010. Основное же ОЗУ остается свободным для работы с прикладными программами.

Компьютерный комплекс выставлялся в де-

## Вам интересно?

Положение с компьютеризацией народного образования общеизвестно. Несмотря на то что еще в 1985 г. был введен курс «Основы информатики и вычислительной техники», до сих пор большинство школ компьютеров не имеют.

Еще хуже обстоит дело с учебными наглядными пособиями для изучения компьютеров. Этим вопросом вообще никто не занимается.

Поэтому в кружке «Компьютер» Киевской городской станции юных техников был разработан, изготовлен и успешно используется в работе набор учебных модулей и узлов компьютера, позволяющий собирать различные структуры микро-ЭВМ и обрабатывать с их помощью учебные программы, написанные на различных языках программирования.

Набор состоит из следующих модулей: микропроцессорного; памяти; ввода-вывода; управления; сопряжения с телевизором; сопряжения с магнитофоном; программатора; блока питания.

Большинство из них имеют несколько модификаций. Например, микропроцессорный модуль изготовлен в таких трех вариантах: простейший модуль, содержащий только БИС типа КР580ИК80;

более сложный модуль, содержащий две микросхемы (КР580ИК80 и КР580ГФ24); еще более сложный модуль, в состав которого входят помимо КР580ИК80 и КР580ГФ24 шинные формирователи К589АП16 и ряд логических микросхем се-

кабре 1988 г. на ВДНХ СССР и был отмечен наградой. Те, кто хочет сейчас, не дожидаясь прекрасного будущего, эффективно использовать вычислительную технику, могут обращаться по адресу: 420008, Казань, ул. Ленина, 18, Казанский государственный университет, ЦИВТ, СП «Диалог».

Ориентировочная стоимость комплекса (БК-0010, магнитофон, монитор, дополнительные платы, программное обеспечение) — около 3000 рублей. К сожалению, своими силами мы можем изготовлять лишь около 80 комплексов в год, и, если какое-либо государственное или кооперативное предприятие заинтересуется нашей разработкой, мы готовы передать ее для более широкого тиражирования.

**В. САМОТУГА**

рии 155 для получения различных управляющих сигналов.

Модуль памяти также имеет разновидности. Изготовлены следующие типы запоминающих устройств:

статическое ОЗУ на К541ПУ2;

динамическое ОЗУ на К565ПУ5;

энергонезависимое ОЗУ на К537ПУ10;

ПЗУ для монитора на К573РФ2;

ПЗУ для интерпретатора Бейсика на К573РФ6.

Помимо учебных модулей разработаны и изготовлены отдельные типовые узлы микро-ЭВМ (триггер, регистр, шинный формирователь и т. д.).

Соединение модулей и узлов друг с другом выполняется при помощи многоконтактных разъемов, установленных на платах (штыревые разъемы) и общей шине (гнездовые разъемы).

Для наглядности на лицевой стороне платы модуля нанесено цветное изображение функциональной схемы модуля, а наиболее ответственные микросхемы расположены в прямоугольниках схемы. Все элементы и соединения, относящиеся к адресному каналу, обозначены синим цветом, к каналу данных — зеленым, к цепям управления — красным. В эти же цвета раскрашены проводники общей шины и соединительные шланги с разъемами.

**В. ТИЩЕНКО,**

руководитель кружка «Компьютер»  
Киевской городской СЮТ

**От редакции.** Подробное описание упомянутых устройств могло бы быть опубликовано в нашем журнале, если это заинтересует достаточное число читателей. Пишите нам!



## ПОЧТОВЫЙ ЯЩИК

Обзоры читательских писем появлялись на страницах нашего журнала примерно раз в год. Сейчас, когда писем приходит все больше, пришла пора сделать выход «Почтового ящика» более регулярным.

*«О необходимости изучения языка программирования. Чем, кроме изучения языка, заниматься в школе, тем более в провинции, когда нет ЭВМ, нет пакетов программ, баз данных и т. п.?»*

*Изучили элементы языка, основные операторы, написали на уроках простенькие программки и пошли (поехали) в соседние учебные заведения, в областной центр. Если повезет, учащиеся сами введут программы; если программы введет оператор ВЦ — тоже лучше, чем ничего. К сожалению, для большинства сельских школ это только мечта.*

*А еще говорят, что БК слишком плохая машина... Нет, БК нужен школе! Пусть БК, пусть что угодно, лишь бы учащиеся видели ЭВМ. И хорошо бы уменьшить поставку в школы полноразмерных КУВТов, а давать комплекты из 2—3 рабочих мест учащихся, зато в большом количестве».*

**Ю. М. ТАМАНИН.**  
Липецкая обл., г. Лебедянь

Предложение о выпуске КУВТ на меньшее количество рабочих мест (может быть, даже на 2—3 места) заслуживает, на наш взгляд, рассмотрения. Это мог бы быть комплекс типа «БК — БК»; на РМП — магнитофон с электронным управлением (он уступает дисководу в скорости, но зато более дешев и доступен). По мере удовлетворения потребностей школ в вычислительной технике такой КУВТ легко можно будет наращивать дополнительными РМУ, а при желании заменить магнитофон дисководом и расширить ОЗУ на РМП.

О том, как это предложение будет воспринято в Гособразовании СССР, мы расскажем в одном из следующих номеров.

*«В нашем кружке радиоэлектроники среди прочего разбирается работа логических элементов и других, более сложных компонентов ЭВМ, изготавливаются действующие демонстрационные макеты этих устройств. Очень жаль, что «ИНФО» редко публикует материалы на эту тему. Ведь такие мате-*

*риалы являются связующим звеном между информатикой, радиоэлектроникой и образованием. Без них информатика получается изолированной, существующей сама по себе.*

*И читателям «ИНФО» нужно быть активнее, помогать своим коллегам. Если есть какие-либо конструкции демонстрационных и наглядных пособий по информатике и основам цифровой вычислительной техники, предложите их редакции журнала. Ведь без наших предложений редакции и публиковать-то будет нечего».*

**А. Г. ПАШКОВ,** г. Рязань **117**

Нам представляется, что для публикаций такого рода больше подходит журнал «Радио», но, если наши читатели активно поддержат вышеприведенные соображения, мы станем уделять этой теме больше внимания. Предлагаем в ответах на публикуемую ниже анкету упомянуть о своем отношении к этой теме.

*«В нашей школе установили КУВТ «Корвет». Есть у него и положительные, и отрицательные стороны. Попробую разложить свой опыт работы с ним «по полочкам».*

*Положительные стороны.*

1. Процессор КР 580 ВМ 80 — аналог Intel 8080 и дает возможность использовать программное обеспечение, накопленное в мире для 8080 (фраза взята из инструкции; мне воспользоваться этим преимуществом не удалось).

2. Мощная версия MSX Бейсика.

3. Развитая локальная сеть.

*Отрицательные стороны.*

1. Дисководы японские, зато ОС наша. После исполнения команды FILES директорий сохраняется в памяти, и можно вынуть диски, можно сменить их — выдаваться будет только список файлов исходного диска. Более того, он может испортить директорий нового диска, если вы попытаетесь работать с ним.

2. В руководстве написано, что программы для «Ямахи» пригодны для «Корвета». Однако у Бейсика «Корвета» отсутствуют операторы DROW (на ученической ПЭВМ), PLAY (хотя микросхема звука есть), INPUT ☒ ( ) (на ученической ПЭВМ), KEY. Есть недостатки в работе операторов READ, INPUT #, функций для работы со строковыми переменными.

3. После двух месяцев работы неисправны уже пять ПЭВМ. Одна была неисправна с самого начала. Проявляется дребезг клавиш.

Есть и кое-что еще, но это уже мелочи. Впрочем, как оценить самопроизвольное изменение кодов клавиш?

Еще раз повторяю, что все описанное — мои личные впечатления и наблюдения».

Р. КАНТОР,

Ленинград, СШ № 393, X класс

Хотелось бы узнать и отношение к поднятым проблемам разработчиков, изготовителей, создателей системного обеспечения, а также пользователей КУВТ «Корвет».

«Необходимо печатать информацию (с ценами) о возможности приобретения зарубежной техники. Ведь многие учебные заведения имеют спонсоров с валютой».

К. Н. СВИДЛЕР, Свердловск

118

Постараемся подобрать и опубликовать такие данные. Мы будем рады, если они пригодятся. Интересно, многие ли могут рассчитывать на валюту? Сообщите нам об этом в ответах на анкету.

А тем, у кого шефы богаты, но не слишком щедры, может помочь цитата из американского журнала «Форчун»: «Американские корпорации имеют вескую причину оказывать помощь в деле повышения уровня профессиональной подготовки. Причина эта — собственные интересы. Сотрудничая с местными учебными заведениями в разработке учебных программ ... безвозмездно передавая в учебные заведения современное оборудование и посылая своих сотрудников работать в качестве преподавателей и наставников, компании стремятся обеспечить свои предприятия лучше обученной и более производительной рабочей силой». Или другая цитата, из лондонского «Экономиста»: «При правительстве Тэтчер политика в области образования была сочтена слишком важным делом, чтобы ее выработкой занималось только министерство образования и науки. Министерство торговли и промышленности финансировало проект оснащения всех школ страны компьютерами».

«...не одну Рапиру у нас игнорируют, наоборот, ее хоть считают учебным языком. А многие языки, используемые за рубежом в качестве учебных, у нас игнорируются. Речь идет о языке Форт. В 1988 г. вышла первая в СССР книга по ФОРТУ, в которой он назван третьим по популярности (за рубежом) после Бейсика и Паскаля.

Я убедился, что это очень мощный и красивый язык, занимающий мало места в памяти и настолько простой, что применяется за рубежом в качестве учебного. Ваш журнал должен помочь распростране-

нию нового для нас и популярного во всем мире языка».

И. М. ГИНЗБУРГ,

БГУ им. В. И. Ленина, I курс

Форт действительно занимает мало места в памяти, но только потому, что изначально это лишь заготовка языка. Все необходимые пользователю операции и функции нужно описывать самостоятельно; а когда они будут описаны, сколько места потребуется ФОРТУ?

С другой стороны, язык этот действительно интересен и гибок. Возможно, мы напечатаем что-нибудь о нем. А пока, может быть, у кого-то из наших читателей уже есть опыт работы с ФОРТОМ, использования его в вузе или школе?

А вот несколько писем, комментария не требующих.

«Я хотел бы увидеть того человека, который сказал, что ОСы серии RT-11, в частности ОСДВК и ФОДОС-2, предназначены для пользователя-непрофессионала. Я бы ему сказал... что он, мягко говоря, ошибается.

Никогда не поверю, что многомодульная (одних драйверов больше двух десятков!), занимающая бездну места на диске и в памяти операционная система может похвастаться при первом знакомстве с ЭВМ. То ли дело MS DOS! Всего три файла. Ее пользователь не должен мучительно генерировать систему, вникать в различия мониторов разных версий; он может форматировать дискету нажатием одной-двух клавиш, а не десятка-полтора».

М. КОЛЕСНИКОВ, Белгород.

«Благодарю и редакцию, и автора И. Гарячука за совет, изложенный в заметке «Маленькие хитрости» (ИНФО 3—89, с. 100). Доработанный магнитофон «Протон-402» не дал еще ни одного сбоя при считывании. Единственный совет всем, работающим с этим магнитофоном: не используйте импортные 90-минутные кассеты — эта пленка чересчур тонка для «Протона».

С. МАКСИМОВ, г. Североморск

«Для питания калькулятора МКШ-2 можно использовать комплект электроснабжения кабинета физики типа КЭФ-10, достаточно поставить на электрические щитки, установленные на ученических столах, розетки, соответствующие вилке МКШ-2, и подвести к ним переменный ток с напряжением 42 В.

Пригодны для этой цели также источник электропитания типа ИПТ-0, 4-220-У4, автотрансформатор для передвижных киноустановок АОСК-0,71У2, селеновый учебный выпрямитель ВС-24м. В последнем случае розетки для МКШ-2 подключаются параллельно вторичной обмотке понижающего трансформатора.

Если понижающие устройства заземлены, то описанные способы питания МКШ-2 отве-



чают требованиям правил техники безопасности».

**А. П. БОРОХОБКО,**

Витебская обл., ч. п. Бешенковичи

#### **BT-АНКЕТА**

Чьей BT вы пользуетесь (своей; шефов; другие варианты)? \_\_\_\_\_

Марка BT, год выпуска. \_\_\_\_\_

Как вы получили BT (через органы народного образования; от шефов; другие варианты)? \_\_\_\_\_

Чем оплачена техника (рублями, валютой)? \_\_\_\_\_

Насколько надежно работает электронная часть BT? \_\_\_\_\_

Насколько надежно работает механическая часть BT (дисководы, принтер)? \_\_\_\_\_

Какова у вас ситуация с ППС (наличие, средства на покупку, доступность)? \_\_\_\_\_

#### **ДОСКА ОБЪЯВЛЕНИЙ**

*«Я хотел бы переписываться с обладателями «Микроши». У меня для этой ПЭВМ есть 42 программы на МЛ. Готов поделиться со всеми желающими».*

**В. В. МУТОВКИН,**

224023, Брест,

ул. Кривошеина, 3/1—24

*«Предлагаю обмен игровыми и системными программами для БК-0010. Прошу помощи в приобретении программных средств для ПЭВМ «Нейрон».*

**В. Б. ГИРУБНИКОВ.**

310115, Харьков,

ул. XVII партсъезда, 10, кв. 100

*«Ищу товарища по интересам, владельца компьютера «Атари», желающего обмениваться программами».*

**А. К. КОЛОДИЙ.**

200032, Таллинн,

бульвар Карла Маркса, 44, кв. 38

## **Научно-методическое объединение «Непрерывное образование»**

### **ПРЕДЛАГАЕТ:**

комплект из 20 высококачественных черно-белых фотографий 18×24 «Технические средства ЭВМ» для оформления стенда, цена — 40 рублей;  
слайд-фильм о профилактике СПИД, цена — 100 рублей;  
слайд-фильм о борьбе с наркоманией, цена — 100 рублей;  
резидентный словарь для IBM-совместимых ПЭВМ объемом 9000 лексических единиц на четырех дискетах 5, 25, цена — 1000 рублей (ДОС).

Для приобретения надо перевести деньги на р/с № 608730 во Фрунзенском отделении ЖСБ г. Москвы;  
указать письменно точный адрес и комплект поставки;  
для словаря выслать 4 дискеты.

### **ПРОИЗВОДИТ**

#### **набор слушателей на двухнедельные курсы:**

- ▲ педагогического мастерства (общение, деловые игры);
- ▲ рационального чтения (продуктивность работы с литературой повышается в 2—3 раза).

Стоимость обучения по каждому курсу — 350 рублей.

Слушатели обеспечиваются гостиницей.

Зачисление на основе гарантийных писем.

**Наш адрес: 125319. Москва, ул. Черняховского, 9.**

**Телефон: 946-38-19.**

## Второй пленум Общесоюзного научно-методического совета информатизации образования

### Решение пленума

1. Принять в целом «Концепцию информатизации образования» с учетом результатов ее доработки на пленуме.

2. Руководящему составу Совета в месячный срок создать рабочую группу из представителей Совета по подготовке проекта «Основные направления работ и фундаментальных исследований в области информатизации народного образования».

3. Руководящему составу Совета организовать рабочую группу, которой поручить в двухмесячный срок подготовить необходимые документы, регламентирующие работу экспертного совета по программно-методическому и техническому обеспечению информатизации образования при Общесоюзном центре НИТ.

4. Одобрить предложение о придании Общесоюзному научно-методическому совету информатизации образования статуса юридического лица со своим расчетным счетом, печатью и другими реквизитами. Принять в целом дополнения к Положению о Совете. Поручить Главному ученому секретарю Совета в месячный срок проработать вопрос с юристами, внести соответствующие изменения в Положение о Совете и представить на утверждение в Гособразование СССР.

5. Главному ученому секретарю Совета оформить соответствующим образом и передать адресату «Обращение участников второго пленума к Верховному Совету СССР».

6. Создать в составе ОНМС НИТ секцию повышения квалификации специалистов.

Пленум проходил в октябре 1989 г. в г. Таганроге на базе Таганрогского радиотехнического института.

Главному ученому секретарю Совета в двухмесячный срок подготовить предложения по составу секции и представить их на утверждение в Гособразование СССР.

7. С целью формирования государственной политики в области информатизации образования на основе национальных стратегий ввести в состав секции координации представителей всех союзных республик.

8. Руководящему составу Совета в месячный срок оформить и разослать «Концепцию информатизации образования» и другие материалы работы второго пленума всем членам Совета и уровневым центрам НИТ.

9. Ввести в практику работы Совета информирование на пленуме членов Совета о проведении важнейших мероприятий, документах Гособразования СССР в области информатизации образования и результатах выполнения ранее принятых Советом решений.

10. Обратить внимание Гособразования СССР на несвоевременное и неполное выполнение решения коллегии Гособразования СССР от 9.12.88 г. № 19/1 «Об организационной структуре системы информатизации образования» в части создания общесоюзного и уровневых центров НИТ.

11. Просить Гособразование СССР:

с целью практического осуществления мер по претворению в жизнь «Концепции информатизации образования» считать недопустимым исключение предмета «Основы информатики и вычислительной техники» из учебного плана школ, ПТУ и техникумов как важнейшего мировоззренческого курса, а также снижение уровня и темпов компьютеризации средних учебных заведений, централизованного оснащения их техникой и программным обеспечением;

разработать механизм материального сти-

мулирования работников народного образования, активно применяющих и обеспечивающих применение НИТ в профессиональной деятельности;

вести в практику работы регулярное информирование на пленумах широкой общественности о планах распределения финансовых и материальных средств, выделяемых на информатизацию образования, их реализации и о достигнутых результатах;

ввиду предстоящего широкого внедрения в народное образование, в том числе в средние школы, вычислительной техники типа IBM/PC и PS считать необходимым оснащать в централизованном порядке педагогические вузы, институты усовершенствования учителей, учебные заведения системы повышения квалификации, а также организации, занимающиеся разработкой инструментальных и программно-методических средств для такого типа ЭВМ из фондов Гособразования СССР, начиная с 1990 г.;

решать проблемы информатизации образования на принципах организации скоординированной на местах работы по созданию конкретных технологий обучения на основе НИТ, ориентированных на реальные потребности и возможности конкретных регионов.

12. Обратить внимание Гособразования СССР, ГКВТИ СССР и министерств-производителей на недопустимо низкую надежность поступающей в учебные заведения ВТ отечественного производства, беспомощность службы обслуживания и ремонта.

13. Просить журнал «Информатика и образование» организовать публикацию информации:

- о различных семинарах, конференциях и совещаниях по данной тематике;
- о государственных ценах на средства вычислительной техники.

## **Обращение пленума в Верховный Совет СССР**

Определяющая роль в осуществлении информатизации общества принадлежит народному образованию.

Мы, участники пленума, выражаем свою озабоченность сохраняющимся положением в деле ресурсного обеспечения народного образования.

В передовых странах мирового сообщества расходы на образование сопоставимы с расходами на оборону. Значительная часть

этих средств направляется на информатизацию образования.

Справедливо декларируемая приоритетность народного образования в нашей стране на практике сегодня не подкреплена приоритетностью его ресурсного обеспечения. Решение задач информатизации образования при этом становится просто невозможным, что обрекает нас на мировую информационную изоляцию уже в ближайшем будущем.

Выделяемые же средства направляются практически только на технику, ее ремонт и обслуживание. Низкая надежность предоставляемых компьютеров приводит к непомерному и неоправданному росту затрат.

Общепризнанный факт большей ресурсоемкости разработок в области программного обеспечения в сравнении с техническим до сих пор не отражается в практике финансирования и расходовании средств, что вызывает их омертвление.

Положение усугубляется тем, что надежды на обеспечение сферы народного образования современными и надежными средствами новых информационных технологий силами отечественной промышленности на сегодня и ближайшую перспективу нереальны.

Решающую роль для осуществления качественного перехода здесь могли бы сыграть:

1. Создание национального фонда народного образования, распределяемого Советом по народному образованию. Создание и финансирование за его счет государственной целевой программы информатизации образования. Формирование национального фонда народного образования с максимальным привлечением возможных источников.

2. Возложение функций генерального заказчика средств новых информационных технологий на Гособразование СССР с выделением соответствующего финансирования без посредников. Целевое выделение средств на разработку программного обеспечения и НИР в области новых информационных технологий.

3. Выделение необходимых валютных ассигнований для закупки всех средств информационной технологии образования, а также разрешение Гособразованию СССР выполнять бартерные операции на основе наукоемкой и малотоннажной продукции, произведенной, в частности, и с привлечением труда учащихся.

Быстрое решение обозначенных проблем позволит создать базу для технологического прорыва в развитии всего народного хозяйства страны.

# Международное совещание в Гатчине

19—23 июня 1989 г. в г. Гатчине под Ленинградом состоялось международное рабочее совещание, на котором разрабатывался прогноз развития исследований по проблеме 1.2.7 «Совершенствование системы образования на основе применения средств вычислительной техники» (профтехобразование) КП НТП СЭВ на 1991—2000 гг. и проект Детализированной программы сотрудничества на 1991—1995 гг.

При разработке прогноза подчеркивалось, что основной задачей на ближайшее десятилетие является внедрение в систему профессионально-технического образования новых информационных технологий, которые реализуются на базе информационной техники: ЭВМ и сети ЭВМ, видеосистемы, спутниковая связь, коммуникационная техника и др.

При рассмотрении ретроспективы и современного состояния научных исследований по проблеме 1.2.7 (профтехобразование) было отмечено, что в 1986—1990 гг. исследования проводились по трем основным направлениям:

1. Разработка технико-педагогических требований к кабинетам вычислительной техники и перспективному комплексу учебной вычислительной техники, создание учебного оборудования на базе ЭВМ и микропроцессоров.

2. Разработка инструментальных и педагогических программных средств для учебных заведений профтехобразования.

3. Разработка рекомендаций и методических документов, определяющих содержание подготовки в области вычислительной техники (учащихся профессионально-технических учебных заведений, студентов инженерно-педагогических институтов и факультетов и слушателей системы повышения квалификации).

Совещание отметило, что наряду с определенными успехами в реализации проблемы 1.2.7 КП НТП СЭВ (профтехобразование) есть и существенные недостатки.

Основным недостатком работы по проблеме 1.2.7 явилась узкая трактовка проблемы исследования. Так, в 1986—1990 гг. не рассматривались вопросы использования таких средств новых информационных технологий, как видеосистемы, сети ЭВМ, средства коммуникаций.

Еще одна проблема — отсутствие аппаратно и программно совместимых средств ВТ, ориентированных на применение в образовании, в странах — членах СЭВ.

Следующая группа проблем обусловлена отсутствием организационной и норматив-

ной основы для внедрения новых информационных технологий в сферу образования, что привело к невозможности выполнения в полном объеме ряда заданий Детализированной программы сотрудничества. Это определяется тем, что в числе организаций-исполнителей в годы текущей пятилетки отсутствовали организации, которые могли бы обеспечить внедрение разработок в практику работы учебных заведений. Принципиальным недостатком научных исследований за истекший период явилось отсутствие общей концепции информатизации образования стран — членов СЭВ и прогноза развития исследований по проблеме 1.2.7 (на период 1986—1995 гг.) — основных стратегических документов.

Проведенный ретроспективный анализ обусловливает основные направления исследований и разработок:

1. Проблемы технического обеспечения процесса внедрения НИТ в образование (исследование возможностей внедрения в образование средств новой информационной техники).

2. Проблемы программного и методического обеспечения внедрения новых информационных технологий в учебно-воспитательный процесс профессионально-технических учебных заведений.

3. Проблемы кадрового обеспечения внедрения НИТ в профессионально-техническое образование (вопросы содержания обучения, подготовки, переподготовки и повышения квалификации).

4. Проблемы создания организационной системы, обеспечивающей возможности использования НИТ.

В итоге состоявшегося обсуждения были определены следующие темы исследований по указанным направлениям на 1991—1995 гг.:

влияние информатизации общества на требования к профессиональной квалификации рабочих;

определение содержания и разработка средств и методики преподавания информатики и основ автоматизации при подготовке квалифицированных рабочих;

актуализация учебного содержания<sup>1</sup>, программных средств и методик подготовки рабочих для использования компьютерных и микропроцессорных средств в профессиях машиностроения, радиоэлектроники и приборостроения, для сферы обслуживания;

исследование дидактических, методиче-

<sup>1</sup> Под актуализацией учебного содержания понимается его изменение и обновление с учетом требований НИТ.

ских и экономических возможностей и границ применения компьютеров как средства обучения в различных странах (сравнительный анализ);

разработка инструментальных программных средств для реализации новых информационных технологий в профессиональном образовании (первоочередная по отношению к собственно разработке педагогических программных средств);

разработка программных средств для совершенствования обучения по общетехническим и специальным дисциплинам для различных направлений профессиональной подготовки;

разработка программного и методического обеспечения интерактивного видео для использования в профессиональной подготовке;

разработка тренажеров на базе микропроцессорной, микрокомпьютерной и видеотехники для подготовки квалифицированных рабочих;

разработка содержания, программных средств, методики подготовки и непрерывного повышения квалификации инженерно-

педагогических кадров в области использования новых информационных технологий в профессиональной подготовке;

разработка систем информационного обеспечения учебного процесса и управления учебным заведением.

Данная тематика исследований отражена и в Детализированной программе сотрудничества на 1991—1995 гг.

Уважаемые читатели! Разработчикам Детализированной программы и прогноза развития исследований было бы интересно узнать ваше мнение о предложенной тематике исследований, а также найти заинтересованных соисполнителей работ по указанным темам. Ваши предложения, замечания и дополнения направляйте по адресу: 191119, Ленинград, ул. Черняховского, 2, ВНИИ профтехобразования, отдел технических и программных средств обучения, лаборатория создания ППС ВТ и применения ЭВМ.

М. ЛЕБЕДЕВА,

канд. пед. наук ВНИИ профтехобразования

123

## Болгаро-советский семинар

По инициативе Центра информационных технологий и информационного обслуживания (ЦИТИО) и кафедры «Автоматизированная обработка социально-экономической информации» Высшего института народного хозяйства им. Д. Благоева (ВИНХ), при содействии Союза научных работников Болгарии (СНРБ) и Общества болгаро-советской дружбы с 7 по 11 мая 1989 г. в Варне был проведен Первый болгаро-советский семинар «Компьютеризация обучения в вузе». Семинар ставил перед собой задачу предоставить возможность для установления прямых контактов между учеными двух стран, работающими по проблеме 5.5 Временной рабочей группы по подготовке кадров в области вычислительной техники и проблеме 1.2.7 Комплексной программы НТП стран — членов СЭВ.

В работе семинара приняли участие более 70 ученых из 20 институтов и организаций СССР и Болгарии, таких, как Московский институт управления, Московский институт железнодорожного транспорта, Высший институт народного хозяйства им. Д. Благоева (Варна), Высший экономический институт им. К. Маркса (София), Ассоциация пользователей персональных ЭВМ (Киев), Ташкентский институт народ-

ного хозяйства, Московский авиационно-технологический институт и др.

На пленарном заседании был представлен доклад С. Дражева «Технологические аспекты разработки программ для компьютерного обучения в экономических вузах». Автор — руководитель темы 5.5.12 «Библиотека программ для компьютерного обучения и научных исследований в экономических вузах» в рамках СЭВ. В докладе рассмотрены возможности макетирования (прототипирования) программного обеспечения для обучения с использованием многооконного интерфейса и автоматического генерирования программ. Наряду с обсуждением доклада была проведена демонстрация библиотеки программ для компьютерного обучения ВИНХ.

Среди докладов, представленных на заседаниях секций, отметим следующие: «Информационно-справочная система по средствам активного обучения» (Ю. Коган, А. Федоровский); «ТУРБО-система для автоматизированного создания компьютерных обучающих курсов» (Ц. Чакалов, Т. Димитрова); «Особенности организации самостоятельной работы студентов по составлению программ модульной структуры» (Д. Дурсунов, М. Гафурова); «АРМ для обучения по дисципли-

лине «Комплексный анализ и планирование промышленности» (И. Иванов, П. Диолова, Т. Атанасова, Е. Градинарова); «Организация компьютерной подготовки инженеров в Белорусском политехническом институте» (Д. Доманевский, В. Павловец, А. Пятунин).

На заключительном заседании было решено сделать семинар постоянно действующим. По предложению участников следующий болгаро-советский семинар намечено провести в октябре 1990 г. в СССР. Был избран организационный комитет, в состав

которого включены М. Гафурова (Ташкент), Р. Георгиева (София), С. Дражев (Варна), А. Кангро (Рига), А. Коган (Киев), В. Толмачев (Минск), А. Цирков (Москва).

Участники семинара имели возможность посетить ЦИТИО при ВИНХ, телевизионную студию, где создаются видеокomпьютерные фильмы для обучения экономистов, и лаборатории института.

Доклады и сообщения будут опубликованы.

Член оргкомитета семинара  
М. ГАФУРОВА

## Ждем продолжения

124

В 1989 г. по инициативе кабинета ИВТ Карагандинского ОИУУ на базе санатория-профилактория «Учитель» был создан областной компьютерный лагерь «Юный программист». Целью этого лагеря было оздоровить и обучить детей из сельской местности (в основном детей работников народного образования Карагандинской области) компьютерной грамотности. На один компьютерный класс в нашей области приходится 20—25 школ. Есть сельские районы, где нет ни одного компьютерного класса.

Из 121 участника 112 впервые сели за компьютер. Во время приема в санаторий-профилакторий было выявлено 92 % детей с теми или иными отклонениями в состоянии здоровья и хроническими заболеваниями. В санаторий на этот период были приняты педиатр, стоматолог, лор-врач. Работали психотерапевт, физиокабинеты, водолечебница, кабинет ЛФК, массажист. Всем детям была проведена витаминизация, фитотерапия по показаниям. В физиокабинете прошли курс лечения 75 человек, водолечебнице — 16 человек, у лор-врача — 55 человек, у стоматолога — 93 человека.

В распоряжении детей были 50 компьютеров: «Агатъ», БК-0010, «Ямахи». Возраст детей от 9 до 16 лет. Они были разбиты на 8 подгрупп по возрастам и занимались по три часа в день: два часа — учебные занятия по заранее составленной программе и один час — компьютерные игры. Играть на компьютерах нравится почти всем. А вот устойчивый интерес к учебным занятиям проявляли не все дети. В процессе наблюдений из детей до 12 лет интерес к учебе проявили 20 %; от 12 до 14 лет — 65—70 %, от 14 и старше — около 80 %. Наибольший интерес к работе с компьютером

ми был у мальчиков. Дети с восторгом занимались на «Ямахах», «Агатах» и отказывались работать на БК-0010 с шильдовой клавиатурой.

В ходе занятий выяснилось, что не хватает программного обеспечения для всех видов компьютеров, особенно для «Агата» с мегабайтным дисководом. Очень мало создано программ для детей младшего возраста. Из-за нехватки компьютеров две подгруппы ежедневно приходилось возить в другой район города. Дети занимались в две смены. Замечено, что во вторую смену дети больше утомлялись, хуже усваивали материал.

По окончании учебы была проведена олимпиада, которая показала, что дети усвоили управление компьютерами, текстовый и графический редакторы, клавиатуру, операторы и синтаксис Бейсика, Рапиры, команды ввода программ на Фокале и Т-языке. Научились составлять простенькие программы и вводить их в компьютер. Первые места заняли дети из Осакаровского, Нурымского, Ульяновского районов Карагандинской области. Дети очень ждут компьютеры в свои школы. Первый областной компьютерный лагерь выявил много способных сельских детей, которых кабинет ИВТ ОИУУ будет в дальнейшем привлекать для участия в олимпиадах по информатике.

Для занятий с детьми привлекались лучшие преподаватели ОИВТ города — А. Б. Ананьева, В. Ф. Казначейский, Е. Ю. Чебакова. Инициатором создания лагеря и его руководителем была методист кабинета ИВТ ОИУУ Л. Койсина.

Часть путевок (20 %) была бесплатной, остальные — стоимостью 15 рублей 30 копеек. Для усиленного питания детей обком

профсоюза выделил дополнительные средства. Дети жили по два человека в комнате с хорошими бытовыми условиями.

Конечно, были и недоработки: не хватало игровых средств, мест для спортивных занятий, не совсем удачно был спланирован

досуг детей. Ведь санаторий существует всего чуть больше полугода.

Однако хочется верить, что карагандинский областной лагерь «Юный программист» будет иметь продолжение, и в процессе становления недостатки будут изжиты.

## **Школьно-студенческое конструкторское бюро МИРЭА**

(ШСКБ МИРЭА) отмечает 10-летний юбилей. К нему ШСКБ разворачивает новое, социально значимое направление: комплекс работ по социальной реабилитации и компьютерному обучению учащихся на базе школы-интерната № 68 и Дома ребенка № 11 Москвы.

Обучение и творческая работа школьников проходят на разных уровнях: от адаптации на игровых компьютерах до разработки интеллектуально углубленных программ, реализуемых в классе IBM PC Московского городского дворца пионеров и школьников.

Задача движения — дать детям профессии, связанные с информатикой, в частности оператора ЭВМ.

Организации, учреждения, центры НТТМ — все, кто хочет принять участие в работах по социальной реабилитации и компьютерному трудовому обучению воспитанников специализированных детских учреждений, — обращайтесь по адресу в межведомственный учебно-методический центр, организованный РК ВЛКСМ и центром НТТМ «Прогресс» Черемушкинского района Москвы (директор С. Л. Серегин).

Адрес: 117297. Москва, ул. Кржижановского, д. 4а, Дом пионеров и школьников Черемушкинского района. Телефон: 125-23-59.

Для тех, кто хочет перечислить средства в фонд этого движения, сообщаем счет комплексного клуба НТТМ «ШСКБ МИРЭА»: № 465305 в Гагаринском отделении Жилсоцбанка Москвы, МФО 20131 или текущий счет № 1700805 в Черемушкинском отделении Жилсоцбанка Москвы, МФО 201490.

## **Ассоциация**

### **«Советская интеллигенция в защиту детства»**

при Детском фонде им. В. И. Ленина, Комитет по делам молодежи при СНИО СССР, передача «Диалог с компьютером» Центрального телевидения ведут подготовительную работу по организации творческого объединения научных работников, изобретателей и детей («Союз творцов»). С этой целью в настоящее время создается база данных «Взрослые — детям». Задача — собрать имеющийся опыт работы с одаренными детьми. Для этого предполагается систематизировать данные о подобных программах при различных государственных и общественных организациях, а также информацию об индивидуальном опыте работы с детьми, который может быть полезен для распространения. Вся информация будет заноситься в память персонального компьютера и может быть использована без ограничения всеми желающими. Создание этой базы данных будет проходить при непосредственном участии школьников.

Предполагается проведение семинаров, дискуссий, кратковременных стажировок как взрослых, так и детей.

Инициативная группа «Союз творцов» предлагает всем желающим принять участие в создании БД «Взрослые — детям», а также быть ее пользователями. С конкретными предложениями просим обращаться по адресу: 117419. Москва, ул. Донская, 37, МАХОТКИНУ Вячеславу Евгеньевичу. Телефон: 234-37-19.

По призыву Общественной ассоциации «Движение за массовое творчество» (Синее движение) в нашей стране ведется большая работа по созданию системы массового творчества. По инициативе Президиума ассоциации в сентябре 1989 г. была создана Академия творчества СССР, объединившая в своих рядах деятелей науки и культуры нашей страны и многих зарубежных государств. Для проведения теоретических исследований в области массового творчества при Союзе научных и инженерных обществ СССР был создан Комитет по проблемам массового творчества. Усилия Общественной ассоциации «Движение за массовое творчество» поддержала международная организация ЮНЕСКО. К решению практических проблем массового творчества активно подключились недавно созданные в Москве информационно-культурный центр Всемирного десятилетия развития культуры и инициативный клубный Центр друзей ЮНЕСКО (ИКЦДЮ).

Пользуясь предоставленной возможностью, мы обращаемся к читателям журнала «Информатика и образование», которые интересуются проблемами творчества и культуры, философии и психологии, педагогики и информатики принять непосредственное участие в деятельности создаваемого при ИКЦДЮ отделения Московского центра массового творчества.

Мы предлагаем всем, у кого есть оригинальные идеи и гипотезы, интересные произведения или инициативы, вопросы или предложения, обращаться письменно по адресу: Москва, 113461, абонементальный ящик 111, ИКЦДЮ. Наиболее интересные материалы будут опубликованы в ежемесячных бюллетенях ИКЦДЮ и в периодической печати, а их авторы будут приглашены для работы в центры массового творчества. Стране нужны ваши идеи. Нам нужны вы. Мы готовы работать вместе. Творчество народа спасет мир!

**А. ПОДЗОРОВ,**

Ученый секретарь Комитета по проблемам массового творчества

**Э. САЛИКОВ,**

исполнительный директор инициативного клубного центра друзей ЮНЕСКО

126

## Нельзя экономить на школах

Экономика США, занимавшая в былые времена несомненно лидирующее положение в мире, сдает позиции. Для исследования причин этого явления в Массачусетском технологическом институте, одном из наиболее известных американских научных центров, была создана Комиссия по производительности труда в промышленности. Проанализировав ситуацию, она выработала рекомендации по преодолению наметившегося отставания США от конкурентов. Далее — цитата из ее выводов.

«Наиболее важной областью долгосрочных капиталовложений является национальная система среднего образования. Более высокий уровень знаний, полученных в школе будущими специа-

листами, станет одним из решающих факторов повышения профессионализма, необходимого для обеспечения роста производительности труда в промышленности. Без существенного повышения качества преподавания в начальной и средней школах никакие меры

регулирования на макроэкономическом уровне или технологические новшества не смогут обеспечить повышение уровня жизни населения».

**ЧТО?**  
**МОЖЕТ?**  
**ЭВМ**

## Однокристалльный «глаз»

Сотрудниками фирмы Sharp Corp. (Япония) создан опытный образец «разумного» формирователя видеосигналов для трехмерных изображений. Это однокристалльная (монокристалльная) система распознавания образов, состоящая из трех слоев. Верхний слой обеспечивает фотоприем и преобразование оптических сигналов в электрические, второй слой преобразует полученную информацию в двоичный код, а третий обеспечивает сравнение данных и связь с устройством отображения.



## Алгоритмический язык — примеры и задачи

Одной из трудностей для любого учителя информатики является подбор задач, особенно по теме «Алгоритмы. Алгоритмический язык». Хорошим подспорьем в этом стали сборники «Задачник-практикум по алгоритмическому языку» для X класса и «Алгоритмический язык: задачи и решения» для XI класса, подготовленные преподавателями Воронежского государственного университета<sup>1</sup>. В них содержится около 500 задач различного уровня сложности.

Большое место авторы отводят задачам на отработку навыков составления и записи алгоритмов. Содержание задач отражает межпредметные связи курса информатики с математикой, физикой, русским языком, историей, географией. Ряд задач требует предварительной формализации, выяснения информационной сущности различных явлений, построения математической модели, что, с одной стороны, способствует усвоению фундаментальных законов информатики, а с другой — наполняет учебные задачи практическим содержанием.

Есть задачи с подробным решением или ука-

<sup>1</sup> Ускова О. Ф., Горбенко О. Д. Задачник-практикум по алгоритмическому языку. Воронеж: Изд-во Воронежского ун-та, 1987.

Ускова О. Ф., Воицкая Г. Э. Алгоритмический язык: задачи и решения. Воронеж: Изд-во Воронежского ун-та, 1989.

занием к решению, некоторые даны без решения и указаний. В конце сборников имеются упражнения для самостоятельной работы. Интересна подборка так называемых обратных задач — по заданному алгоритму определить, какую задачу он описывает. Задачи сгруппированы по принципам методической (формирование умений и навыков) и логической (содержательной) взаимосвязи.

Сборники предназначены для широкого круга читателей. Преподаватели, студенты, учащиеся школ найдут в них задачи по силам и по душе. Хотя приводимые решения записаны на школьном алгоритмическом языке, их с успехом можно записать и в подавляющем числе примеров исполнить на алгоритмическом языке или языке программирования. Поэтому они будут полезны учителям, придерживающимся различных методических подходов и использующим разные учебные пособия. Авторы не навязывают конкретной методики обучения решению задач — перед учителем открыты широкие творческие возможности.

Некоторые задачи доступны и учащимся VI—VIII классов, что позволит использовать книги в случае перевода курса ОИВТ в средние классы.

Читателю, взявшему в руки эти полезные книги, предстоит увлекательное путешествие в мир алгоритмов. Хочется пожелать ему успехов.

В. ФРЕЙМАН

127

## Через задачи — к программированию

Арсенал задач учителей информатики пополнился серией содержательных и увлекательных игровых, математических и других задач, представленных в книге «Через задачи — к программированию»\*. Разбор решения, рассмотрение соответствующего математического аппарата, применение неожиданных способов машинной реализации известных или новых задач позволяют показать учащимся практическое применение математических знаний и приобщить их к искусству программирования. Заслуживает внимания тот факт, что предлагаемые задачи были реализованы в практике работы Малой академии наук школьников Крыма «Искатель». Программы написаны на одной из версий Бейсика. Наряду с подробным обсуждением алгоритмов и пояснениями соответствующих компьютерных программ читателю предлагаются краткие исторические справки и порою любопытные сведения.

Традиционно «скучная» по изложению тема о системах счисления становится интересной за счет удачного подбора задач, например подсчитать количество цифр в числе 100, реализовать дешифра-

тор. Достаточно внимания уделено задачам о замечательных числах (Фибоначчи, Бернулли, Каталана, Мерсенна и т. д.).

Обсуждение возможностей машинной графики проводится на примере задачи на построение прямоугольного треугольника по двум заданным медианам, проведенным к катетам. При этом разбираются разные способы решения: подход геометра и подход математика-программиста. Привлекает внимание использование ЭВМ для рисования геометрических мультфильмов. Реализация геометрических идей с помощью ЭВМ осуществляется при конструировании экранных геометрических инструментов: экранной линейки, экранного циркуля, экранного эллипсостроителя и др.

Учащимся предлагается попробовать свои силы в разработке программного обеспечения простейшей информационно-справочной системы.

Реализация алгоритмов игр, создание игровой программы, целый ряд занимательных задач: «магический квадрат», задание «Кванта», «ЭВМ, поработайте с кроссвордом» и многие другие, несомненно, вызовут интерес.

Книга поможет учителю обеспечить действенные связи между информатикой и математикой, продемонстрировать удивительные возможности применения ЭВМ.

И. АНТИПОВ,

НИИ школ МНО РСФСР

\* Касаткин В. Н. Через задачи — к программированию: Для ст. шк. возраста. Киев: Радянська школа, 1989. 128 с. (Серия «Когда сделаны уроки»).

## Общедоступный словарь по вычислительной технике

Микропроцессор — блок в составе устройства ЦВМ, к-рый управляет работой устройства... в соответствии с командой центрального процессора ЦВМ.

*Советский энциклопедический словарь. М.: Советская энциклопедия, 1983*

Повсеместные успехи в деле компьютеризации нашего общества вступили в резкое противоречие с уровнем компьютерной так называемой грамотности, что сделало необходимость издания подобного словаря настоятельной. Ориентация на самые широкие круги читателей заставила автора основное внимание уделить доступности, по возможности даже общедоступности пояснений. Задача эта затруднялась тем, что специфические компьютерные термины зачастую маскируются под самые обычные слова, что многих сбивает с толку. Чрезвычайно важно понимать, что, например, «поддержка» в информатике означает не совсем то, что поддержка в фигурном катании, и совсем не то, что поддержка политики партии и правительства.

На наш взгляд, автору удалось преодолеть все трудности. Мы с удовольствием представляем читателю его многотомный труд в журнальном варианте.

*Адресное пространство* — пространство, включающее в себя поле памяти, куда имеет доступ процессор. Огорожено специальными перемычками, которые можно переставлять. Процессор перебирает на ножках адреса необходимые адреса и так перебирается в любую точку пространства (т. е. забирается в любую ячейку памяти).

*Аналого-цифровой преобразователь (АЦП)* — специальное устройство, подключаемое к аналоговой вычислительной машине с целью преобразования ее в цифровую ЭВМ, если у пользователя возникла в этом необходимость. Современные АЦП обладают высоким быстродействием.

*Видеоконтроллер* — устройство в составе большинства современных ЭВМ. Позволяет пользователю не только видеть, но и контролировать содержимое скрытого от него под кожухом оборудования ЭВМ. В. гораздо эффективнее, чем вентиляционные отверстия.

*ДИП-корпус.* а) Разновидность корпусов микросхем в виде сороконожек. Выведена за границу, к нам завезена сотрудниками дипломатического корпуса (отсюда название). б) По другим данным, эту разновидность впервые вывел наш соотечественник задолго до появления ее на Западе. Названием стала аббревиатура слов «догнать и перегнать».

*Контроллер прямого доступа* — специализированное устройство в составе ЭВМ (как правило, выполняется в виде крупной микросхемы — жучка с большим количеством мощных ножек), гарантирующее пользователю проворный доступ

к любой ячейке посредством расталкивания стоящих на пути других компонентов: транзисторов, микросхем, конденсаторов и даже, как показали испытания, тяжелых трансформаторов.

*Модулятор / демодулятор (модем)* — устройство, подключаемое к ЭВМ в случае необходимости осуществить ее разбиение на отдельные модули, а также для последующего синтеза полученных модулей в единое целое. Служит для локализации неисправностей.

*Оперативное запоминающее устройство (ОЗУ)* — быстродействующее запоминающее устройство для хранения оперативной информации, которую оно теряет при отключении питания. Постоянная угроза в любую наносекунду потерять данные заставляет его работать очень быстро.

*Операционная система* — совокупность служебных программ, состоящих из отдельных операторов, оперативно оперирующая данными. Вместе с аппаратной поддержкой в виде четырех ножек, на которые она опирается, образует компьютер.

*Процессор* — одно из центральных устройств в структуре ЭВМ, выполняет обработку данных ему данных. По важности уступает единственно-му блоку в машинной иерархии — цессору.

*Персональная ЭВМ* — ЭВМ, предназначенная для персонала того или иного предприятия и используемая в следующих целях:

- 1) выполнение компьютерных игр (наиболее подходящи импортные модели ПЭВМ);
- 2) выполнение ее ремонта (обычно используются отечественные модели);
- 3) удовлетворение радилюбительского влечения, заключающееся в поисках компонентов для ПЭВМ и ее частичной сборки.

*РИСК-архитектура* — архитектура процессоров, применяемая разработчиками в странах Запада, при которой игнорируются элементарные нормы безопасности пользователей в погоне за максимальной прибылью монополий. Экономия получается за счет упрощения системы команд, а также аппаратных средств.

*Система приоритетного обслуживания* — комплекс мер, принимаемых в рамках той или иной вычислительной системы в целях наиболее справедливого обслуживания узлов и блоков самой системы. Перечень услуг включает: питание, охлаждение, профилактический осмотр; реже — диагностику, отправку на заслуженный отдых.

Пример: дешифратор, обладающий дворянским титулом, обслуживается по более высокой категории, чем простой шифратор.

*Средства отладки* — совокупность программно-аппаратных средств для наладки и пуска ЭВМ. Достаточно дороги и малодоступны. В связи с этим большинство специалистов используют сначала средства отладки процесса наладки, а исчерпав их, довольствуются средствами отладки причин неполадки, которые очень дешевы и развивают интуицию.

Внимание ПТУ, техникумов, школ с углубленным изучением  
вычислительной техники

Омский приборостроительный завод им. Козицкого приступил к серийному выпуску

**универсальных лабораторных стенов,**  
предназначенных для проведения лабораторных работ по курсу «Основы автоматизации и вычислительной техники».

Стенд позволяет проводить лабораторные работы по изучению и исследованию: основных логических элементов (10 типов), триггеров (5 типов), регистров (4 типа), различных комбинационных устройств (преобразователей кодов, дешифраторов, мультиплексоров, демультиплексоров, сумматоров, арифметическо-логического устройства); **оперативного запоминающего устройства; операционного блока процессора; модели ЭВМ.**

Режимы работы и входная информация задаются с пульта управления.

Индикация данных осуществляется в двоичном, десятичном и шестнадцатиричном кодах с помощью светодиодных индикаторов и на цифровом дисплее. В состав стенда входят основной блок, 6 сменных плат и набор из 27 сменных трафаретов со схемами исследуемых узлов.

*В комплект поставки входят методические рекомендации по использованию стенда в учебном процессе.*

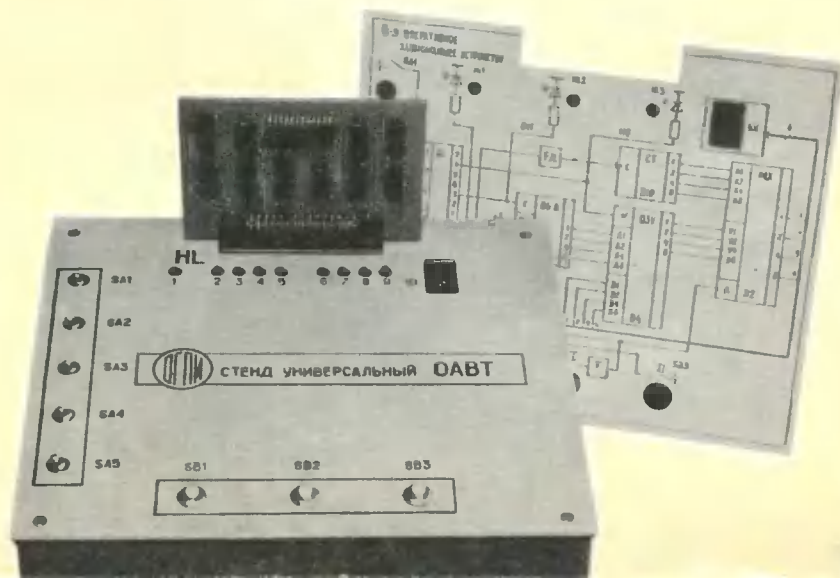
Габариты 245×200×100 мм.

Стенд выпускается в двух вариантах: с питанием от источника постоянного тока 5 В; 0,7 А и с питанием от сети.

Цена первого варианта — 300 рублей, второго — 370 рублей.

Гарантируется поставка в 1990 г.

*Заявки направлять по адресу:*  
644099, Омск-99, Приборостроительный завод им. Козицкого.



Б.Д.И. 11-41

60 коп.  
Индекс 70423

OldPC.ru  
7003  
музей компьютеров

ИНФ  
1'90



ИНФОРМАТИКА  
И ОБРАЗОВАНИЕ

