

ИНФОРМАТИКА И ОБРАЗОВАНИЕ

1991





ИНФОРМАТИКА И ОБРАЗОВАНИЕ

Содержание

Методика обучения

- Фрейман В. Методические подходы к обучению школьников работе с базами данных 3
Степанов М. Возможности многостраничной графики 7
Буцик В. Обучение младших школьников началам информатики 16
Плеухова Л. Структура и содержание мотивационного обеспечения обучающих программ 20

Кабинет ВТ

- Алексеев М. Пролог-Д на «Агате» 25
Соловьев Ю. Несколько слов о «Роботроне-1715» 29
Комаров С., Мониин М. Про Бейсик, быстродействие и кое-что еще 32
Соболевский М. Гармоничная учеба 36
Лучко О., Дьяченко А. Программно-аппаратная система «Светофор» 39
Большаков Е., Перцовский Н., Трушников А. Учебная система управления базами данных 41
В продаже — МС 1502 42

Клуб БК

- Зальцман Ю. Архитектура и ассемблер БК 43
Диков А., Калашников А., Кулаков А. TermOS 49
Оптимальный драйвер магнитофона для БК 52
Подключение принтеров «Электроника МС-6312» и «Электроника МС-6313» к БК 55
Программатор микросхем ППЗУ 558РРЗ для БК-0010 57
Увеличение символов на экране 59
Волшебная сила двух ячеек 60
Кирпичики ваших программ 61
О символьных переменных 62
Спрайты на БК 63
Рекомендации по подключению внешних устройств к порту ввода—вывода БК-0010 64
Важные восьмеричные числа 65

Внеклассная работа

- Прохоров В., Раков С., Кириухин В. Олимпиадные задачи по информатике 67
Случай в городе Глунове 71
Касаткин В., Переход И. Задача о сумме факториалов 75

Молодежная инициатива

НИТ в дошкольном образовании

Марголис Я., Иванов А. Шестилетки: к творчеству через компьютер	85
Белавина И. Психологические последствия компьютеризации детской игры	91
Вы нам писали	94
Педагогические кадры	
Козлов О. Поэтапная подготовка учителей-предметников к проведению занятий в кабинете информатики	97
Точка зрения	
Кирюхин В. Всесоюзные олимпиады по информатике — вчера, сегодня, завтра	101
Федотов В. Учебные программы в стихии рынка	105
Зарубежный опыт	
Полат Е., Литвинова А. Информационные технологии в зарубежной школе	109
Нам пишут	
Компьютеры в обучении: учит ли история?	115
О физической неточности	116
Как бы есть	117
Информация	
Выставка программных средств	119
Обращение к учителям информатики	123
Наука, искусство и компьютеры	124

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)

Редактор отдела *А. Кравцова*
Научный редактор *Н. Копытина*
Зав. редакцией *Н. Игнатова*
Художественный редактор *Л. Коновалова*
Корректор *М. Суворова*

Сдано в набор 22.03.91. Подписано в печать 26.04.91. Формат 70×100 1/16.
Бумага офсетная № 1. Печать офсетная. Усл. печ. л. 10,40. Усл. кр.-отт.
42,88. Уч.-изд. л. 13,48. Тираж 56 295 экз. Заказ 445. Цена 1 руб. 20 коп.

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

Почту направлять по адресу: 119034, Москва, Смоленский б-р, д. 4.
Издательство «Педагогика». Журнал «Информатика и образование».
Адрес редакции: Лефортовский пер., д. 8.
Т е л е ф о н: 261-11-29.

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

© «Педагогика», «Информатика и образование», 1991

В. ФРЕЙМАН

Методические подходы к обучению школьников работе с базами данных

В статье [1] мы рассмотрели опыт проведения практических занятий по информатике, на которых была выявлена возможность обучения школьников работе с реальными автоматизированными информационными системами, основанными на использовании баз данных в качестве информационного фонда. Проведенное в 1988—1990 гг. исследование показало более широкие перспективы организации целостного системного изучения баз данных учащимися средних школ. Актуальность связанных с этим методических вопросов подтверждают последующие публикации в журнале по концептуальным проблемам развития методической системы школьной информатики [2, 3], а также по вопросам изучения школьниками баз данных и информационно-поисковых систем [4, 5].

В данной статье рассматриваются аспекты методики обучения школьников работе с базами данных. Эти подходы предложены в нашем исследовании [6].

С методической точки зрения, мы считаем целесообразным исходить из понимания базы данных как целостного научного понятия, обозначающего представление взаимосвязанных данных, характеризующих информационные свойства и связи объектов реального мира в памяти ЭВМ в форме модели соответствующей предметной области с целью эффективного централизованного управления ими в процессе решения прикладных задач.

Соотнося это положение с многообразием типов информационных систем и средств баз данных, мы обнаружили наличие общего способа деятельности пользователя при решении типовых ин-

формационных задач, инвариантного относительно этого многообразия и заключающегося в последовательном применении методов:

анализа информационной потребности пользователей и предметной области базы данных;

синтеза процедур обработки данных в режимах поиска обновления, защиты, преобразования данных;

машинной реализации полученных процедур на компьютере;

интерпретации полученных результатов.

Исходя из этого, целью обучения школьников работе с базами данных является овладение ими указанным общим способом деятельности посредством грамотного применения перечисленных методов при решении учебных задач.

Анализируя различные аспекты деятельности пользователя при работе с базой данных, мы установили, что все они связаны с решением информационных задач двух основных типов:

получение информации на основе данных, уже хранящихся в базе;

создание новой базы данных и поддержание полученной модели предметной области в определенном состоянии.

Именно такого рода задачи мы выбрали в качестве учебных и разработали на этой основе конкретную методику обучения.

В процессе разработки методики было установлено, что общими составляющими элементами решения каждого типа задач являются: *запрос* как форма выражения информационной потребности пользователя, *операции* как средство реализации информационной потребности, *данные* как специфическая

форма представления информации об объекте реального мира в ЭВМ, *технология* как процедура реализации информационной потребности, *результат выполнения запроса*.

Все пять элементов взаимосвязаны — пользователь формулирует запрос на обработку данных об объектах реального мира; для выполнения запроса на ЭВМ необходимо описать совокупность операций, с помощью которых требуемые данные обрабатываются программами СУБД, и выполнить указанные операции в соответствии с технологией решения задачи на ЭВМ, получив результат, адекватный поставленному запросу.

4 Связующие элементы действия — *сформулировать, описать, выполнить, проверить* — тесно связаны с рассмотренными выше общими методами решения информационной задачи и определяют этапы ее решения.

Различные методические приемы, направленные на формирование соответствующих умений у учащихся, основаны на выявлении учащимися общих элементов в условиях конкретной учебной задачи, с одной стороны, и поиске отличий каждой следующей учебной задачи от предшествующей, с другой, с акцентированием внимания школьников на определение способов действий в возникших с появлением этих отличий новых ситуациях.

Остановимся на методических особенностях задач каждого типа. Эти особенности показаны нами на примерах, взятых из классических учебных пособий по базам данных. В конкретной методике с учетом используемых программных средств и предметного наполнения базы данных учитель сам может подобрать аналогичные примеры.

Задача. Тип I.

Дана база данных:

Таблица 1. Поставщики

П№	ИМЯ	ГОРОД	СТАТУС
П1	Смит	Нью-Йорк	20
П2	Джонс	Финикс	10
П3	Блейк	Туксон	30
П4	Кларк	Монреаль	20
П5	Адамс	Торонто	30

Таблица 2. Детали

Д№	НАЗВАНИЕ	ЦВЕТ	ВЕС
Д1	Болт	Красный	12
Д2	Винт	Зеленый	17
Д3	Шайба	Синий	12
Д4	Гайка	Красный	19

Таблица 3. Поставки

П№	Д№	КОЛИЧЕСТВО
П1	Д1	3
П1	Д2	2
П1	Д5	4
П2	Д1	3
П3	Д5	2
П4	Д5	4
П5	Д6	5

Запрос 1. Какие поставщики (имя, город) [7] поставляют детали весом более 12?

Запрос 2. Подсчитать количество деталей каждого цвета, поставляемых каждым поставщиком. Результаты представить в виде таблицы:

Таблица

Поставщик	Количество деталей		
	красных	зеленых	синих

Первый запрос относится к частному случаю задачи использования готовой базы данных: поиску в базе без преобразования данных.

Решение содержащейся в этом запросе задачи как учебной требует такой организации учебной деятельности, при которой учащиеся выполняют следующие действия.

1. Выделяют объекты реального мира, информация о которых хранится в базе данных (ПОСТАВЩИКИ, ДЕТАЛИ), связи этих объектов (поставщик осуществляет ПОСТАВКУ деталей), свойства объектов (имя поставщика, город, статус, название детали, цвет, вес, количество деталей каждого типа, поставляемых каждым поставщиком).

2. Анализируют условие поиска: выделяют объекты, свойства, связи, исследуемые в данном конкретном запросе.

Дано:

ВЕС 12

Найти:

ИМЯ, ГОРОД

3. Составляют ориентировочную основу действий.

3.1. Стратегия поиска: найти детали, обладающие свойством: ВЕС 12, найти номер ПОСТАВ-

ЩИКА, поставляющего эти номера деталей, найти ИМЯ, ГОРОД поставщика с найденным номером. 3.2. Описание поиска (текст запроса на языке используемой СУБД). 3.3. Ввод описания в ЭВМ. 3.4. Выполнение запроса. 3.5. Получение результата. 3.6. Интерпретация результата.

Таким образом, учащиеся «открывают» связь различных элементов выполнения информационного запроса: *запрос — данные — операции — технология — результат*, а также приобретают навыки решения задач данного типа, независимо от применяемых программных средств и предметного наполнения базы данных. В соответствии с психологической концепцией развивающего обучения (В. В. Давыдов) можно утверждать, что в этом случае учащиеся овладевают обобщенными способами действий.

Рассмотрим выполнение школьниками запроса 2.

При попытке решения задачи известными способами учащиеся сталкиваются со следующими препятствиями:

1. В таблице дано количество деталей, поставляемых каждым поставщиком. Для определенного вида (номера) детали известен цвет. Требуется получить данные о новом свойстве поставщика — о количестве деталей одного цвета, поставляемых конкретным поставщиком. Чтобы это узнать, необходимо преобразовать исходные данные. Следовательно, наряду с операцией поиска требуется овладеть операцией преобразования данных (суммой). Здесь учащиеся знакомятся с другим частным случаем задачи первого типа: поиском в готовой базе данных с преобразованием.

2. Результат выполнения запроса необходимо представить в виде таблицы. Для этого нужно «научить» машину заполнять строки и графы данной таблицы (и, конечно, рисовать шапку этой таблицы). После того как с помощью учителя учащиеся знакомятся с генератором отчетов, который обычно входит в состав программных средств работы с базой данных, учащиеся преодолевают и это препятствие. Они са-

мостоятельно описывают общий способ решения задачи:

1. Аналогично 1 в запросе 1

2. Дано:

Поставщик	Номер,	Имя
Деталь	Номер,	Цвет
Поставка	Номер поставщика,	Номер детали, Количество

Найти:

Поставщик, Количество деталей красного цвета, Количество деталей зеленого цвета, Количество деталей синего цвета.

3. Ориентировочная основа действий:

3.1. Разработка эскиза формы отчета. 3.2. Разработка правила заполнения формы данными.

3.2.1. В таблицу поставок вписываем новую графу — цвет, учитывая для каждой детали соответствующий ей цвет по табл. 2.

3.2.2. Для каждого поставщика вычисляем сумму поставок деталей данного цвета.

3.3. Описание формы отчета и правила заполнения формы.

3.4. Ввод описаний в ЭВМ. 3.5. Выполнение операций с помощью компьютера. 3.6. Интерпретация полученных результатов.

При решении этой задачи общий план действий не зависит от конкретных данных, программных, языковых и технических средств.

Рассмотрим пример задач, требующих создания и поддержания новой базы данных.

Задача. Тип II.

Спроектировать автоматизированную систему учета студентов университета, которая должна эффективно выдавать ответы и сообщения по запросам следующего характера: а) составить список всех студентов, изучающих данный курс; б) составить список всех преподавателей и курсов, которые они постоянно читают для данного отделения; в) составить список всех курсов для данного отделения; г) определить, имеет ли студент соответствующую подготовку для изучения данного курса; д) включить студента в число (исключить из числа) слушателей данного курса; е) составить расписание зачета для каждого студента [8].

Первым действием учащихся при ознакомлении с этим типом задач является выделение уже известных им элементов, используемых при работе с готовой базой данных: запроса, данных, операций, технологии, результата.

Анализируя условие задачи, учащиеся определяют, что запросы в неявной форме сформулированы в п. а) — е) задания и выражают информационную потребность пользователя.

Кроме того, учащиеся должны установить наличие еще одного запроса, представленного в условии неявным образом: организовать базу данных и ее поддержание. В свою очередь, этот запрос распадается на несколько частных запросов: создание базы данных, использование базы данных, модификацию базы данных, защиту данных.

Создание базы данных предполагает проведение учащимися анализа предметной области, выделение объектов и их информационных свойств. Далее разрабатываются формы входных документов и описывается их структура по типу запроса 2 (тип I). После этого описание структуры вводится в ЭВМ и производится машинное проектирование структуры базы данных. Затем данные вводятся в базу; правильность исходных данных при этом проверяется по специальным правилам контроля.

Использование базы данных осуществляется по запросам типа а) — г), е) или по разовым запросам, аналогичным рассмотренным выше.

Модификация базы данных проводится школьниками на основе анализа динамики изменения предметной области и может включать: добавление или удаление данных — по типу запроса д), а также изменение структуры исходных документов.

Защита данных предполагает проведение комплекса действий, связанных с обеспечением сохранности информационного фонда и направленных на его защиту от аппаратных и программных сбоев ЭВМ, а также от неправильной организации работы пользователя с базой данных и несанкционированного доступа.

Эксперименты, проведенные в школе (XI класс) и в УПК, показали, что

процесс обучения решению рассмотренных типов учебных задач требует дифференцированного подхода к учащимся. Направления возможной реализации такого методического подхода выявлены нами в результате сопоставления успешности обучения различных групп учащихся с различными требованиями к умениям пользователей баз данных, определяемыми классификацией пользователей.

Первую группу (11 %) составили школьники, плохо усваивающие весь учебный материал, связанный с изучением баз данных. В будущем это вероятные косвенные пользователи баз данных.

Во вторую группу (18 %) попали школьники, наиболее успешно справляющиеся с решением задач первого типа (запрос 1). Это прямые пользователи, использующие приложения. В принципе для знакомства с базами данных им достаточно сведений о базах данных, содержащихся в соответствующих учебниках информатики.

Третью, наиболее многочисленную группу образуют учащиеся (51 %), которые успешно решают задачи первого типа (запросы 1 и 2). Их можно отнести к прямым пользователям, разрабатывающим и использующим приложения. Для них сегодняшний вариант курса информатики в XI классе является недостаточно насыщенным сведениями по базам данных.

Четвертая группа (13 %) очень интересна для учителей, ведущих углубленные занятия по информатике с изучением различных типов информационных систем. Представители этой группы успешно решают реальные производственные задачи (тип II) по созданию небольших информационных систем. По классификации — это прямые пользователи, разрабатывающие и использующие приложения, непрограммирующие.

Учащиеся первых четырех групп являются конечными пользователями ЭВМ.

Не попавшие ни в одну из этих групп школьники (а их 7 %) образуют пятую группу. Как правило, стихией этих ребят является увлечение программированием. Изучение простых, с точки зрения программистов, языков запросов

и описания документов такие школьники считают для себя «деквалификацией». По классификации пользователей они образуют группу потенциальных системных и прикладных программистов — в будущем разработчиков программных средств для конечных пользователей. Для этих учащихся нужен специальный курс информатики, ориентированный на изучение программирования. В нашем эксперименте этот курс они изучали в УПК.

Таким образом, основные методические подходы к организации обучения школьников работе с базами данных заключаются в обучении школьников базам данных как целостному понятию информатики, лежащему в основе разработки и использования автоматизированных информационных систем; в обеспечении овладения учащимися общим способом деятельности при работе конечного пользователя ЭВМ с базой данных; в построении конкретной методики обучения, направленной на усвоение школьниками общих методов решения типовых информационных задач создания и использования баз данных и информационных систем и в адаптации этой методики к условиям реального учебного процесса изучения информатики в школе с учетом требований дифференцированного подхода к учащимся.

Реализация этих методических подходов на практике позволит дать учащимся более широкое понятие о методах изучения информационных свойств объектов реального мира, способах отображения этих объектов и связей

между ними в базах данных информационных систем, а также сформировать у них умения создавать базы данных на ЭВМ и использовать их, извлекая из них необходимую информацию, пополнять их новой информацией, изменять имеющуюся, представлять эту информацию в наглядной форме (например, в виде таблиц). Все это вместе взятое способствует достижению основных целей преподавания информатики в школе.

Л и т е р а т у р а

1. Фрейман В. Базы данных на уроках информатики // Информатика и образование. 1988. № 6.

2. Уваров А. Информатика в школе: вчера, сегодня, завтра // Информатика и образование. 1990. № 4.

3. Кузнецов А., Долматов В. Методическая система обучения ОИВТ: структура и функции, состояние и перспективы // Информатика и образование. 1989. № 1.

4. Бургин М., Степенко Г. Информационный поиск и компьютерная грамотность // Информатика и образование. 1990. № 1.

5. Гриценко А. Информационно-поисковые системы для КУВТ-86 // Информатика и образование. 1989. № 6.

6. Фрейман В. Методика обучения школьников использованию баз данных. Автореферат диссертации на соискание степени кандидата педагогических наук. Москва. 1991 г.

7. Озкарахан Э. Машины баз данных и управление базами данных / Пер. с англ. М.: Мир, 1989.

8. Тиори Т., Фрай Дж. Проектирование структур баз данных: В 2 кн. Кн. 1 / Пер. с англ.; Под ред. В. И. Скворцова. М.: Мир, 1985.

М. Степанов,
НИИ школ МНО РСФСР

Возможности многостраничной графики

Ряд персональных компьютеров, например «Агат» и «Ямаха» в стандарте MSX-2, предоставляют пользователю средства многостраничной графики. Суть их состоит в том, что память или видеопамять компьютера разбита на несколько областей, называемых экранными страницами. Каждая из них содержит

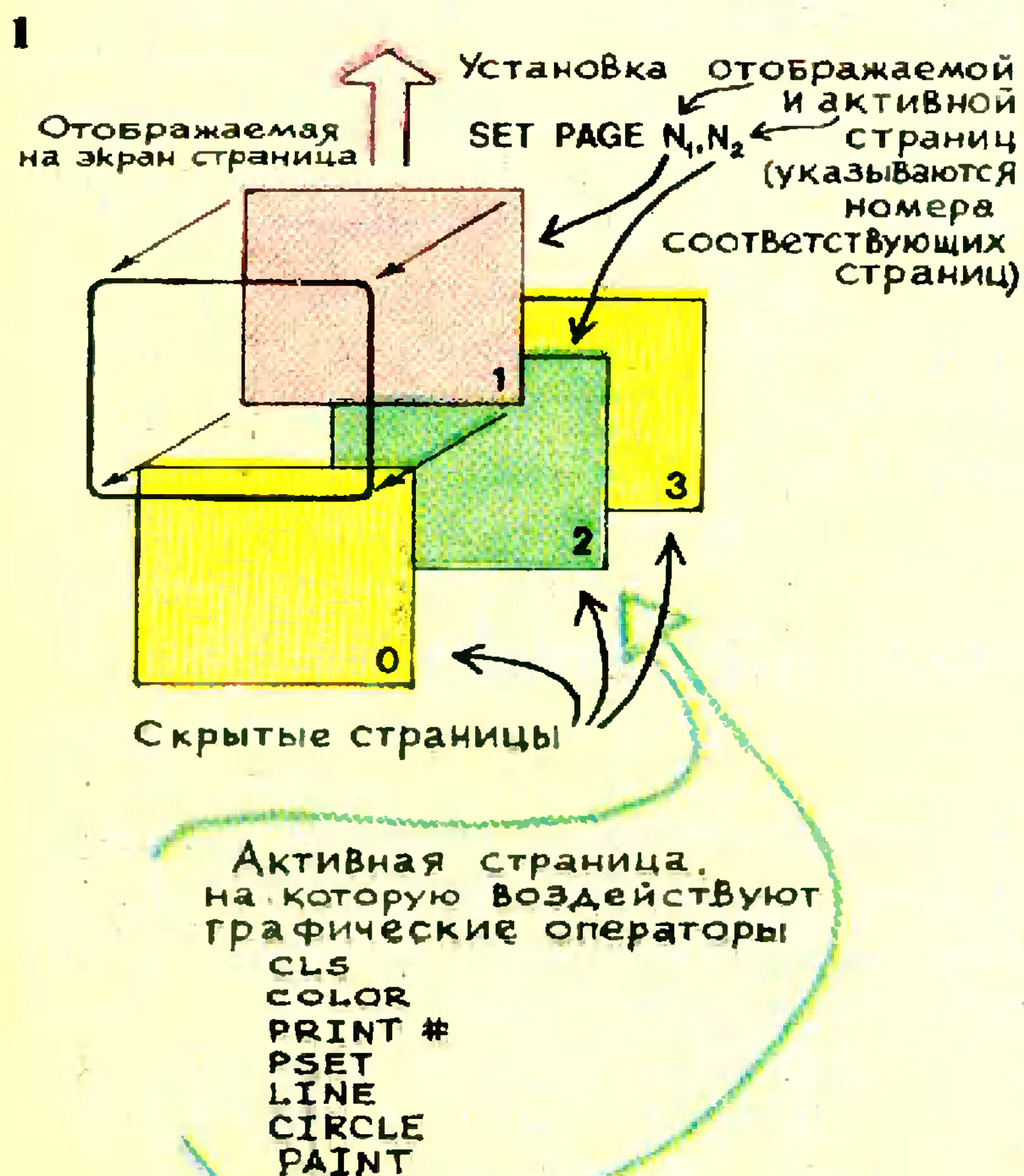
информацию, достаточную для того, чтобы задать на экране изображение в графическом режиме. Пользователь по собственному выбору может отображать на экран любую из экранных страниц и тем самым быстро менять графическое изображение на экране.

Ниже описываются возможности

использования многостраничной графики, проиллюстрированные программами на языке MSX-Бейсик для ПЭВМ «Ямаха». Получение аналогичных результатов на ПЭВМ «Агат» более затруднено, так как предполагает у пользователя знание архитектуры машины и основ ассемблера.

Для чтения данной статьи необходимо знакомство с командами MSX-Бейсика, позволяющими работать в графическом режиме SCREEN 2, общем для компьютеров «Ямаха» в стандартах MSX-1 и MSX-2.

8 Многостраничная графика связана с графическим режимом SCREEN 5 (размер экрана 256×212 точек). В распоряжении пользователя находятся 4 экранные страницы с номерами от 0 до 3. Одна из этих страниц отображается на экран. Еще одна — объявляется активной. Графические операторы, используемые в программе, действуют только на активную страницу. Таким образом, в случае, когда отображаемая и активная страницы не совпадают, действие графических операторов скрыто от пользователя. Номера отображаемой и активной страниц задаются оператором SET PAGE (рис. 1). Этот оператор служит переключателем экранных страниц. Следует помнить, что при переключении страницы не очищаются.



Пример 1. На каждой из четырех экранных страниц нарисуем одну фазу вращения двух пирамид в противоположных направлениях.

```

10 PI=4*ATN(1)
20 SCREEN 5
30 FOR I=0 TO 3
40 SET PAGE I, I:CLS
50 GOSUB 1000
60 NEXT I
70 FOR I=0 TO 3
80 SET PAGE I, I
90 FOR T=1 TO 200
100 NEXT T, I
110 GOTO 70
1000 U1=PI*I/8+.3
1010 U2=U1+PI/2
1020 U3=PI/2-U1
1030 U4=U3-PI/2
1040 X1=120+100*COS(U1)
1050 Y1=100-50*SIN(U1)
1060 X2=120+100*COS(U2)
1070 Y2=100-50*SIN(U2)
1080 X3=120+50*COS(U3)
1090 Y3=100-25*SIN(U3)
1100 X4=120+50*COS(U4)
1110 Y4=100-25*SIN(U4)
1120 LINE(120,210)-(X1,Y1),13
1130 LINE(120,210)-(X2,Y2),13
1140 LINE(120,210)-(240-X1,200-Y1),13
1150 LINE(120,210)-(240-X2,200-Y2),13
1160 LINE(X1,Y1)-(X2,Y2),6
1170 LINE-(240-X1,200-Y1),6
1180 LINE-(240-X2,200-Y2),6
1190 LINE-(X1,Y1),6
1200 PAINT(120,100),6
1210 LINE(X3,Y3)-(X4,Y4),2
1220 LINE-(240-X3,200-Y3),2
1230 LINE-(240-X4,200-Y4),2
1240 LINE-(X3,Y3),2
1250 PAINT(120,100),2
1260 LINE(120,0)-(X3,Y3),10
1270 LINE(120,0)-(X4,Y4),10
1280 LINE(120,0)-(240-X3,200-Y3),10
1290 LINE(120,0)-(240-X4,200-Y4),10
1300 RETURN

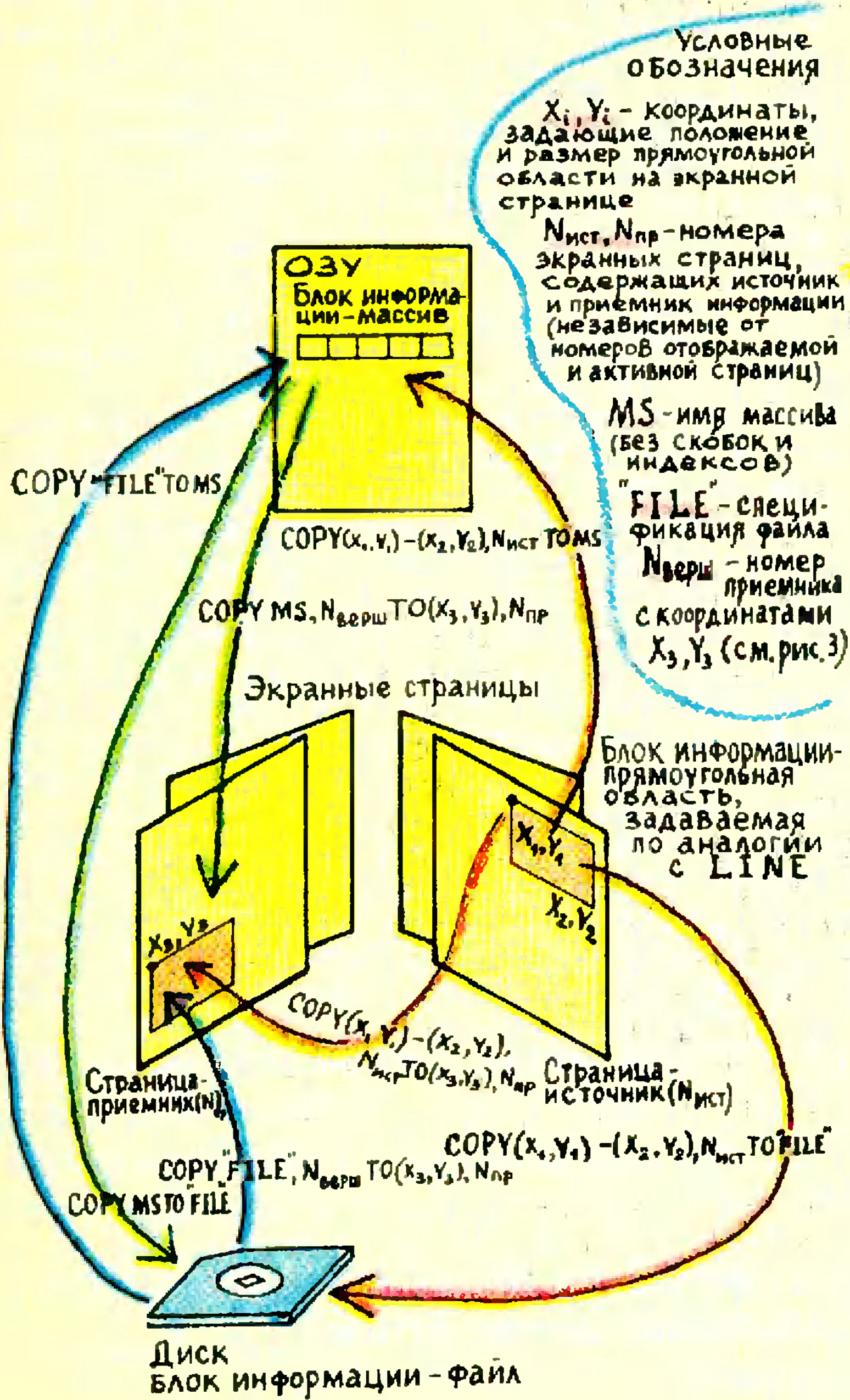
```

В программных строках 30—60 производится последовательное построение фаз движения. Пользователь видит процесс построения на экране, поскольку оператор SET PAGE в строке 40 задает один и тот же номер для отображаемой и активной страниц. В строках 70—110 производится циклическое переключение отображаемых страниц. Цикл с заголовком в строке 90 введен для того, чтобы выдерживать небольшую паузу между последовательными кадрами. Существенным недостатком данной программы является малое количество кадров, не позволяющее получить на экране изображение движущихся объектов с высоким качеством.

Кроме команды SET PAGE важным средством работы с многостраничной

графикой является команда COPY, которая служит для перенесения блоков информации из одних областей памяти в другие (рис. 2).

2



На рисунке указаны основные форматы команды COPY. Отметим, что электронный диск в этой команде использовать нельзя. При пересылке блока информации, задающего изображение в прямоугольной экранной странице в массив, необходимо определить размерность этого массива и описать его. Если стороны прямоугольника содержат соответственно A и B экранных точек, то размерность массива определяется по формуле

$$R = \text{INT}((4 \times (A+1) \times (B+1) + 7) / 8) + 4$$

Наиболее простой, но важной для преподавателя информатики, сферой применения этой команды является ускоренное построение семейств сложных геометрических фигур. Достаточно однажды построить нужную фигуру (предполагается, что построение требует значительного времени), а затем в цикле

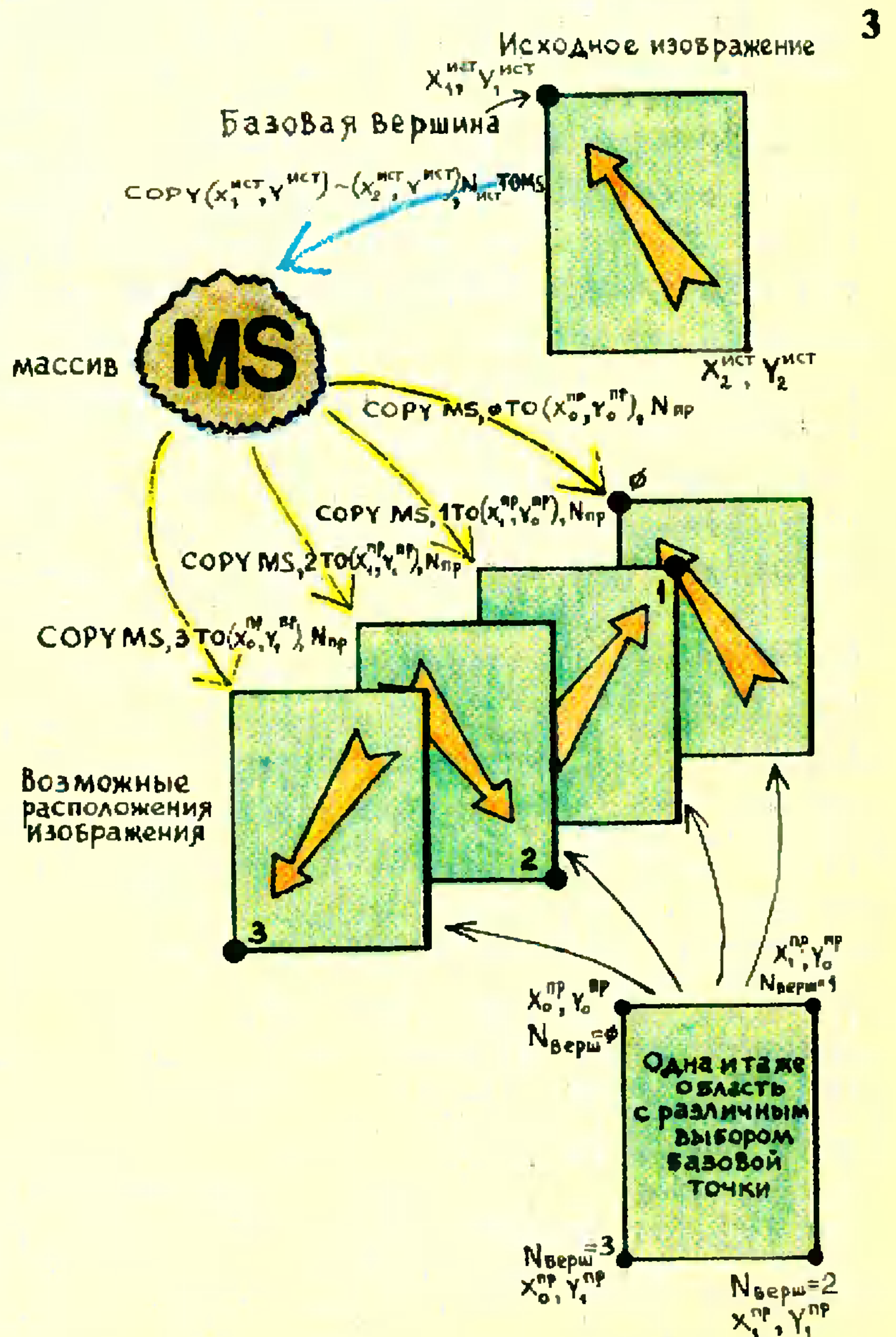
быстро копировать ее на отображаемую экранную страницу.

Пример 2. Построение семейства закрашенных семиугольных звездочек.

```

10 SCREEN 5
20 SET PAGE 0,0:CLS
30 PI=4*ATN(1):R1=14:R2=25
40 PSET(32,1)
50 FOR I=1 TO 14
60 X=32+R1*SIN(I*PI/7)
70 Y=26-R1*COS(I*PI/7)
80 LINE-(X,Y)
90 SWAP R1,R2
100 NEXT I
110 PAINT(32,26)
120 FOR Y=0 TO 52*3 STEP 52
130 FOR X=0 TO 64*3 STEP 64
140 COPY(0,0)-STEP(63,51),0 TO(X,Y),0
150 NEXT X,Y
160 GOTO 160
  
```

Перенесение блока информации, хранящегося в массиве или файле, на прямоугольную область экранной страницы может быть произведено неоднозначно. Параметр $N_{верш}$ (рис. 2), принимающий значения от 0 до 3, указывает, какая из вершин прямоугольной области принимается за базовую (рис. 3).



9

3

Мы будем использовать эту особенность команды COPY для получения симметрично отраженных изображений. С этой целью на экранной странице строится исходное изображение, затем копируется в массив и из массива вновь переносится на экранную страницу с указанием нужного значения параметра $N_{\text{верш}}$.

Пример 3. Скопируем исходное изображение в массив, а затем перенесем его последовательно в четыре прямоугольные области, окружающие точку с фиксированными координатами. Для получения изображения в очередной области нужно нажать пробел.

```

10 SCREEN 5
20 R=INT((4*(20+1)*(40+1)+7)/8)+4
30 DIM FG(R)
40 SET PAGE 3,3
50 CLS
60 GOSUB 1000
70 COPY (0,0)-(20,40),3 TO FG
80 SET PAGE 0,0:CLS
90 FOR I=0 TO 3
100 COPY FG,I TO (100,80),0
110 F$=INPUT$(1)
120 NEXT I
130 GOTO 130
1000 *ФИГУРКА
1010 CIRCLE(0,0),40,,2
1020 PAINT(0,0)
1030 RETURN

```

Пример 4. По аналогии с примером 1 заставим циклически отображаться на экран страницы с заранее подготовленным изображением. При этом, однако, заменим крупные фигуры на семейство мелких (фигурки гимнастов). Подготовку экранных страниц, в том числе симметричное отображение исходных фигурок для получения новых поз, проведем с помощью команды COPY.

```

10 SCREEN 5
20 FOR I=0 TO 3
30 SET PAGE 0,I:CLS
40 NEXT I
50 SET PAGE 2,2
60 GOSUB 1000
70 SET PAGE 1,1
80 GOSUB 2000
90 R=INT((4*(40+1)*(41+1)+7)/8)+4
100 DIM FG(R)
110 COPY (0,0)-(40,41),2 TO FG
120 COPY FG,2 TO (0,41),0
130 COPY (0,0)-(40,41),1 TO FG
140 COPY FG,1 TO (40,0),3
150 FOR I=0 TO 3
160 SET PAGE I,I
170 FOR X=0 TO 205 STEP 41
180 FOR Y=0 TO 150 STEP 50
190 COPY (0,0)-(40,41),I TO (X,Y),I
200 NEXT Y,X,I

```

```

210 FOR I=0 TO 3
220 SET PAGE I
230 FOR T=1 TO 150:NEXT T
240 NEXT I
250 GOTO 210
1000 *ФИГУРА 1
1010 CIRCLE(20,36),5
1020 PAINT(20,36)
1030 LINE(17,30)-STEP(6,-15),,BF
1040 LINE(17,15)-STEP(2,-15),,BF
1050 LINE(21,15)-STEP(2,-15),,BF
1060 LINE(15,0)-STEP(1,1),,BF
1070 LINE(24,0)-STEP(1,1),,BF
1080 LINE(17,30)-STEP(-12,12)
1090 LINE(17,29)-STEP(-12,12)
1100 LINE(17,28)-STEP(-12,12)
1110 LINE(23,30)-STEP(12,12)
1120 LINE(23,29)-STEP(12,12)
1130 LINE(23,28)-STEP(12,12)
1140 RETURN
2000 *ФИГУРА 2
2010 CIRCLE(32,23),5
2020 PAINT(32,23)
2030 LINE(10,20)-STEP(15,6),,BF
2040 LINE(10,26)-STEP(2,15),,BF
2050 LINE(13,40)-STEP(1,1),,BF
2060 LINE(0,10)-STEP(9,9)
2070 LINE(0,11)-STEP(9,9)
2080 LINE(0,12)-STEP(9,9)
2090 LINE(0,13)-STEP(9,9)
2100 LINE(0,9)-STEP(1,0)
2110 LINE(0,9)-STEP(1,1)
2120 LINE(24,8)-STEP(2,29),,BF
2130 RETURN

```

Познакомившись с командами SET PAGE и COPY, мы можем перейти к описанию нескольких стандартных приемов использования этих команд.

Получение движения на экране с помощью смены готовых кадров

1. Построение особого рисунка на каждой из экранных страниц и последовательное отображение этих страниц на экран командой SET PAGE (примеры 1 и 4).

2. Использование фиксированной страницы в качестве экрана. Остальные страницы предназначены для скрытого хранения прямоугольных кадров (как правило, небольшого размера). Перенесение изображения на экран производится командой COPY.

Пример 5. На странице 1 построим 16 кадров, представляющих собой фазы вращения параллелепипеда. Копируя их на фиксированный участок страницы 0, получим на экране изображение вращающегося тела.

```

10 SCREEN 5
20 FOR I=0 TO 1
30 SET PAGE I,I:CLS

```

```

40 NEXT I
50 FOR J=0 TO 3
60 FOR I=0 TO 3
70 X0=(2*I+1)*32
80 Y0=(2*J+1)*25-10
90 U=(J*4+I)*ATN(1)/8
100 X1=X0-30*COS(U)
110 Y1=Y0+15*SIN(U)
120 X2=X0-30*COS(U+3.14159/2)
130 Y2=Y0+15*SIN(U+3.14159/2)
140 GOSUB 230
150 NEXT I,J
160 SET PAGE 0,0
170 FOR J=0 TO 3
180 FOR I=0 TO 3
190 COPY (64*I,50*J)-STEP(63,49),1
                                TO (96,75),0
200 FOR T=1 TO 50:NEXT T
210 NEXT I,J
220 GOTO 170
230 *ПОСТРОЕНИЕ КАДРА
240 LINE(X1,Y1)-(X2,Y2)
250 LINE-(2*X0-X1,2*Y0-Y1)
260 LINE-(2*X0-X2,2*Y0-Y2)
270 LINE-(X1,Y1)
280 PAINT(X0,Y0)
290 LINE(X1,Y1)-(X1,Y1+20)
300 LINE-(X2,Y2+20)
310 LINE-(X2,Y2)
320 IF J>1 THEN 360
330 LINE(X2,Y2+20)-(2*X0-X1,2*Y0-Y1+20)
340 LINE-(2*X0-X1,2*Y0-Y1)
350 RETURN
360 LINE(2*X0-X2,2*Y0-Y2)-
    (2*X0-X2,2*Y0-Y2+20)
370 LINE-(X1,Y1+20)
380 RETURN

```

Внеся небольшие изменения в программу:

```

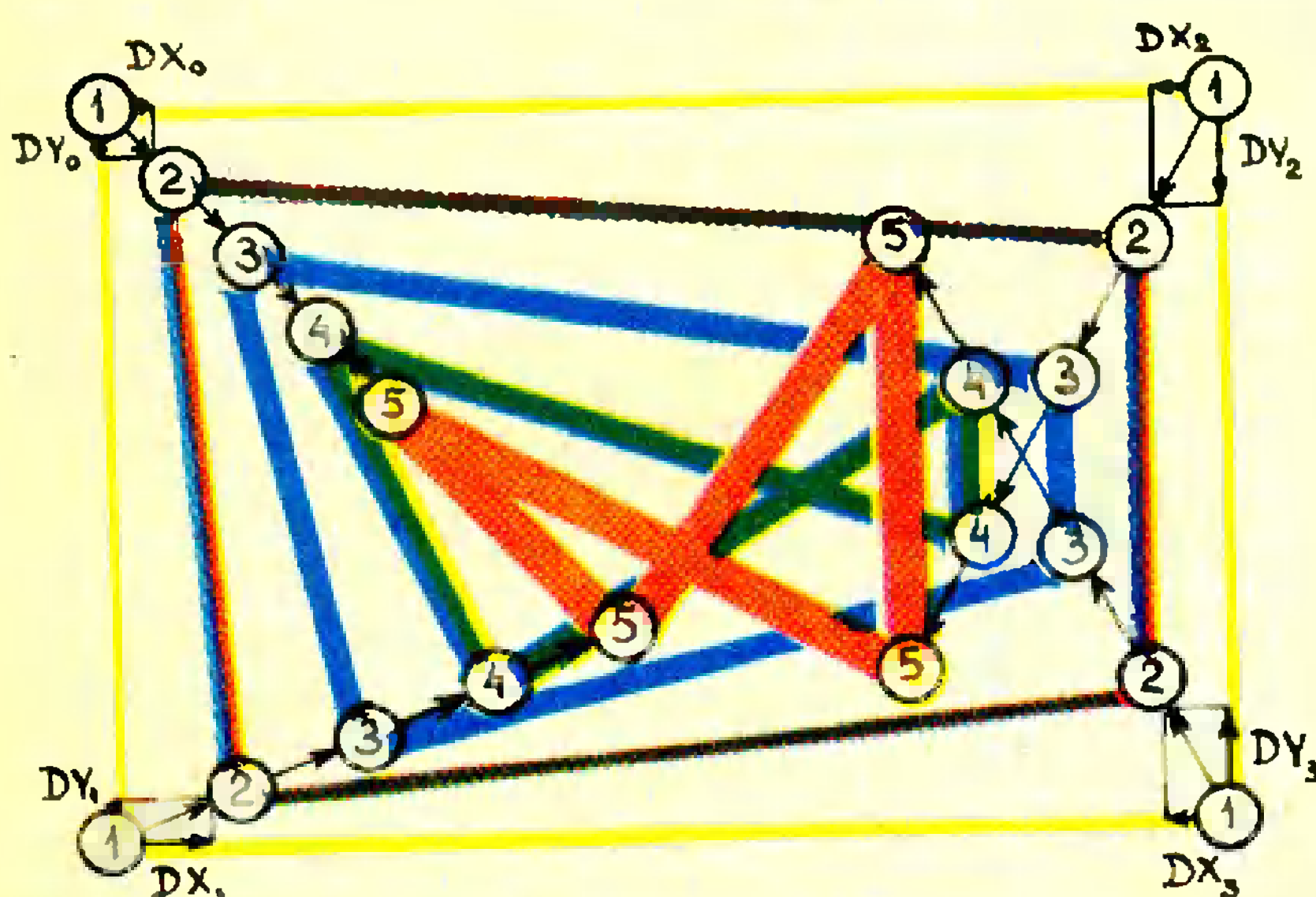
190 COPY (64*I,50*J)-STEP(63,49),1
                                TO (60,75),0
195 COPY (64*(3-I),50*(3-J))-
    STEP(63,49),1 TO (140,75),0

```

заставим вращаться на экране два тела.

Пример 6. Заготовим кадры на трех скрытых страницах. Программа продемонстрирует деформацию прямоугольных рамок в седло и обратную деформацию (рис. 4).

4 Движение вершин деформируемых рамок



```

10 DIM X(3),Y(3),DX(3),DY(3)
20 DATA 1,1,4,-1,-4,8,-1,-9
30 FOR I=0 TO 3
40 READ DX(I),DY(I)
50 NEXT I
60 SCREEN 5
70 FOR S=1 TO 3
80 SET PAGE S,S:CLS
90 FOR NY=0 TO 1
100 FOR NX=0 TO 1
110 NK=(S-1)*4+NY*2+NX
120 FOR I1=0 TO 1
130 FOR I2=0 TO 1
140 I=I1*2+I2
150 X(I)=NX*128+NK*DX(I)+127*I1
160 Y(I)=NY*105+NK*DY(I)+105*I2
170 NEXT I2,I1
180 GOSUB 1000:*ПОСТРОЕНИЕ СЕДЛА
190 NEXT NX,NY,S
200 SET PAGE 0,0
210 SB=1:SE=3:SD=1
220 YB=0:YE=1:YD=1
230 XB=0:XE=1:XD=1
240 FOR S=SB TO SE STEP SD
250 FOR NY=YB TO YE STEP YD
260 FOR NX=XB TO XE STEP XD
270 XT=NX*128
280 YT=NY*106
290 COPY(XT,YT)-STEP(127,105),S
                                TO (0,0),0
300 COPY(XT,YT)-STEP(127,105),S
                                TO (128,106),0
310 NEXT NX,NY,S
320 SWAP SB,SE:SD=-SD
330 SWAP YB,YE:YD=-YD
340 SWAP XB,XE:XD=-XD
350 GOTO 240
1000 FOR J=1 TO 2
1010 FOR K=0 TO 1 STEP 1/16
1020 XA=X(0):YA=Y(0):XB=X(1):YB=Y(1)
1030 GOSUB 2000:FSET(XM,YM)
1040 XA=X(2):YA=Y(2):XB=X(3):YB=Y(3)
1050 GOSUB 2000:LINE-(XM,YM)
1060 NEXT K
1070 SWAP X(1),X(2):SWAP Y(1),Y(2)
1080 NEXT J
1090 RETURN
2000 XM=XA+(XB-XA)*K
2010 YM=YA+(YB-YA)*K
2020 RETURN

```

11

Рассмотрим вопросы, связанные с использованием готовых кадров. Подготовка кадров может проводиться многими путями, например с помощью графического редактора. Однако наиболее удобный путь автоматического построения кадров состоит в том, что выбирается семейство геометрических фигур или линий, определяемое числовым параметром. Выбрав некоторую совокупность значений параметра, можно в цикле построить соответствующие каждому значению конфигурации, которые и являются кадрами, задающими эволюции геометрической фигуры.

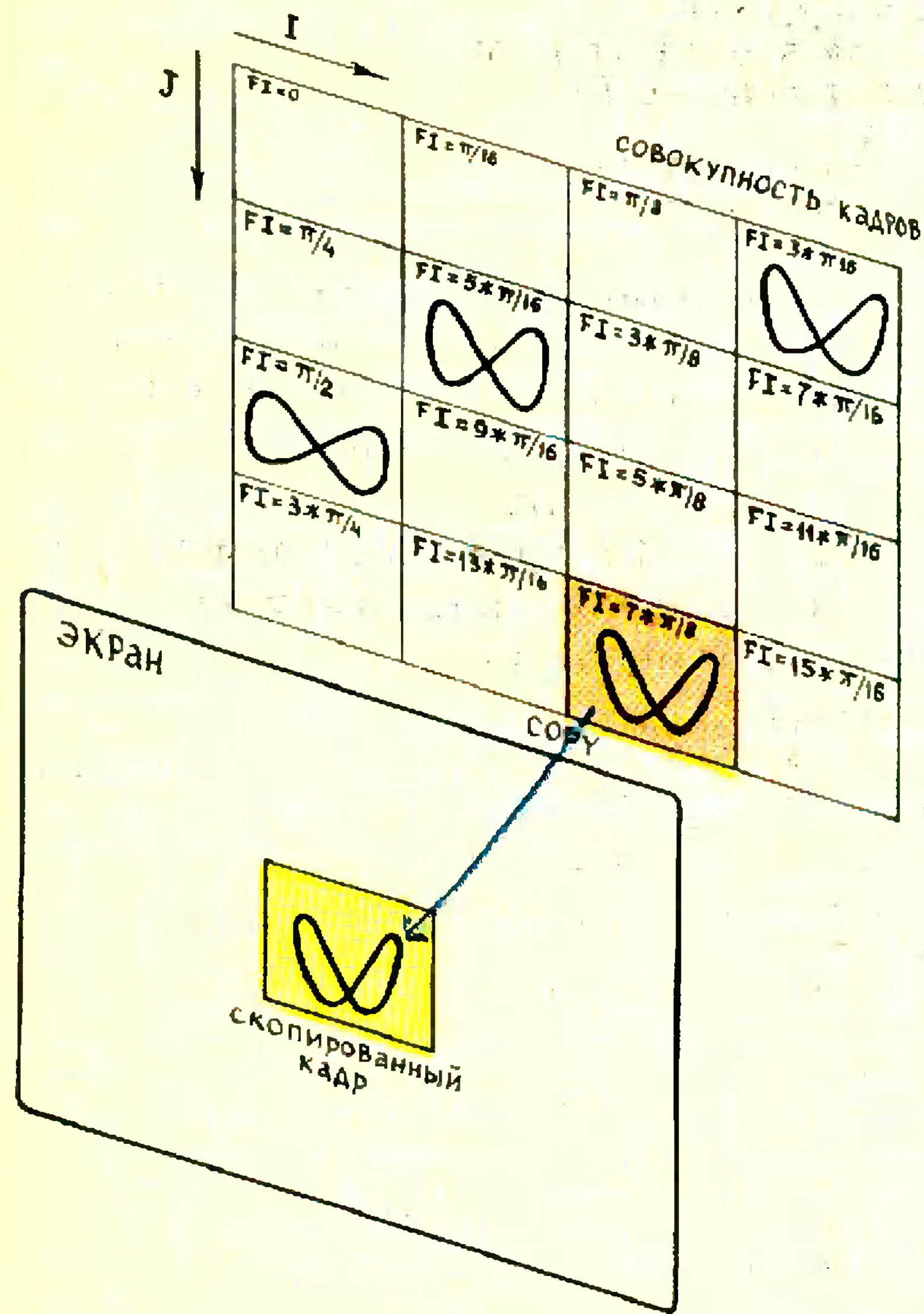
Пример 7. Семейство фигур Лиссажу задается параметром FI по формулам

$$X = A + R1 * \sin(U + FI)$$

U от 0 до 2π

$$Y = B + R2 * \cos(2 * U)$$

Свяжем значение параметра FI с номерами кадров (рис. 5) по формуле $FI = \pi * (4 * J + I) / 16$ и для каждого значения FI построим фигуру Лиссажу.



Если предполагается многократное использование данной программы, то лучше разбить ее на две части. Одна производит построение кадров с последующей записью на диск в файл «KADR» и выполняется один раз.

```

10 SCREEN 5:PI=4*ATN(1)
20 SET PAGE 0,0:CLS
30 FOR J=0 TO 3
40 FOR I=0 TO 3
50 FOR U=0 TO 2*PI STEP .03
60 XT=I*64+32*XSIN(U+PI*(4*J+I)/16)+32
70 YT=J*52+26*XSIN(2*U+PI/3)+26
80 PSET(XT,YT)
90 NEXT U,I,J
100 COPY(0,0)-(255,211),0 TO "KADR"
110 END

```

Вторая производит загрузку кадров с диска (что намного быстрее, чем процесс построения) и копирует кадры на экран.

```

10 SCREEN 5
20 SET PAGE 1,1
30 COPY "KADR",0 TO (0,0),1
40 SET PAGE 0,0:CLS
50 FOR J=0 TO 3
60 FOR I=0 TO 3
70 COPY(64*I,52*J)-STEP(63,51),1
      TO (96,78),0
80 FOR T=1 TO 50:NEXT T
90 NEXT I,J
100 GOTO 50

```

При использовании готовых кадров полезно иметь в виду следующие обстоятельства.

1. Выбор того или иного порядка копирования готовых кадров на экран позволяет получать различные эффекты при одних и тех же кадрах.

2. На странице, которую пользователь выбрал на роль экрана, может быть выделено несколько областей для копирования в них кадров. Кроме того, на этой же странице можно расположить рисунок-декорацию, позволяющий придать целостное восприятие совокупности кадров.

Пример 8.

```

10 DATA AB,AB,A,GB,GB,G
20 DATA FB,FB,F,G,A
30 DATA B-1
40 DATA B-B,B-B,B-,AB,AB,A
50 DATA GB,GB,G,F,G
60 DATA A1
70 DATA AB,AB,A,GB,GB,G
80 DATA FB,FB,F,E,F
90 DATA G1
100 DATA GB,GB,G,FB,FB,F
110 DATA E,E,F,E
120 DATA D1
130 SCREEN 5
140 SET PAGE 0,0:CLS
150 CIRCLE(128,105),100
160 CIRCLE(128,105),101

```

```

10 SCREEN 5:PI=4*ATN(1)
20 FOR I=0 TO 1
30 SET PAGE I,I:CLS
40 NEXT I
50 FOR J=0 TO 3
60 FOR I=0 TO 3
70 FOR U=0 TO 2*PI STEP .03
80 XT=I*64+32*XSIN(U+PI*(4*J+I)/16)+32
90 YT=J*52+26*XSIN(2*U+PI/3)+26
100 PSET(XT,YT)
110 NEXT U,I,J
120 SET PAGE 0,0
130 FOR J=0 TO 3
140 FOR I=0 TO 3
150 COPY(64*I,52*J)-STEP(63,51),1
      TO (96,78),0
160 FOR T=1 TO 50:NEXT T
170 NEXT I,J
180 GOTO 130

```

Построение кадров в данном случае занимает довольно большое время. После того как кадры подготовлены, программу можно прервать и запустить сразу со строки 120 командой прямого диалога:

```
SCREEN 5: GOTO 120
```

```

170 CIRCLE(128,110),20
180 PI=4*ATN(1):U=0
190 SET PAGE 1,1:CLS
200 FOR I=0 TO 7
210 FOR J=0 TO 5
220 X=I*32+16:Y=J*32+16
230 CIRCLE(X,Y),15
240 XZ=X+5*SIN(U)
250 YZ=Y+5*COS(U)
260 CIRCLE(XZ,YZ),10
270 CIRCLE(XZ,YZ),5
280 PAINT(XZ,YZ)
290 U=U+PI/24
300 NEXT J,I
310 SET PAGE 2,2:CLS
320 K=.1
330 FOR J=0 TO 3
340 FOR I=0 TO 3
350 X=I*64+32:Y=J*51+25
360 CIRCLE(X,Y),25,,,K
370 CIRCLE(X,Y),24,,,K
380 K=K+.1
390 NEXT I,J
400 SET PAGE 0,0
410 IG=0:JG=0:N=0
420 BR=0:ER=3:IR=0:JR=0:D=1
430 XG=IG*32:YG=JG*32
440 XR=IR*64:YR=JR*51
450 COPY (XG,YG)-STEP(31,31),1
      TO (58,58),0
460 COPY (XG,YG)-STEP(31,31),1
      TO (164,58),0
470 COPY (XR,YR)-STEP(63,50),2
      TO (96,132),0
480 IG=IG+1:IF IG=8 THEN IG=0:
      JG=JG+1:IF JG=6 THEN JG=0
490 IR=INT(4*RND(1))
500 JR=INT(4*RND(1))
510 READ NT$:PLAY NT$
520 N=N+1:IF N=47 THEN N=0:RESTORE
530 FOR T=1 TO 50:NEXT T
540 GOTO 430

```

Двухэкранная организация движения с прорисовкой на скрытом кадре

В ряде случаев использование готовых кадров неудобно либо невозможно. Например, при изменении размеров объекта в больших пределах. В этом случае приходится рисовать объект, стирать и снова рисовать. В графическом режиме SCREEN 2 можно следующим образом организовать рост окружности:

```

10 SCREEN 2
20 FOR R=5 TO 90 STEP 5
30 CLS
40 CIRCLE(128,96),R
50 NEXT R

```

Качество изображения будет несравненно лучше, если в SCREEN 5 использовать два попеременно включаемых экрана. Пока пользователь смотрит на изображение, построенное на одной странице, на второй — производится скрытое построение новой фазы движе-

ния. Затем выполняется смена состояний обеих страниц: отображаемая становится активной, а активная — отображаемой.

Пример 9. Рост окружности реализуется следующим образом.

```

10 SCREEN 5
20 K0=0:K1=1
30 FOR R=5 TO 90 STEP 5
40 SET PAGE K0,K1
50 CLS
60 CIRCLE(128,96),R
70 SWAP K0,K1
80 NEXT R

```

Указанный способ не исключает возможности копирования готового изображения на два экрана. Выбор копирования или прорисовки зависит от конкретной задачи.

Пример 10. Вращение прямой (объект большой, но быстро рисуемый), используется прорисовка.

```

10 SCREEN 5
20 SET PAGE 0,0:CLS
30 SET PAGE 1,1:CLS
40 K0=0:K1=1
50 FOR X=0 TO 255
60 SET PAGE K0,K1
70 CLS
80 LINE(X,0)-(255-X,211)
90 SWAP K0,K1
100 NEXT X
110 FOR Y=1 TO 210
120 SET PAGE K0,K1
130 CLS
140 LINE(255,Y)-(0,211-Y)
150 SWAP K0,K1
160 NEXT Y
170 GOTO 50

```

Пример 11. Управление с помощью клавиш-стрелок положением кольца на экране (объект рисуется медленно и вписывается в прямоугольную область), используется копирование.

```

10 SCREEN 5
20 FOR I=0 TO 2
30 SET PAGE 0,I:CLS
40 NEXT I
50 R1=30:R2=20:X=128:Y=106
60 CIRCLE(R1,R1),R1
70 CIRCLE(R1,R1),R2
80 PAINT(R1,R1-R2-1)
90 K0=0:K1=1
100 SET PAGE K0,K0
110 COPY (0,0)-STEP(2*R1,2*R1),2
      TO (X-R1,Y-R1),K0
120 SET PAGE K0,K1:CLS
130 S=STICK(0)
140 IF S=0 THEN 130
150 IF (S=8 OR S=1 OR S=2)
      AND Y>R1 THEN Y=Y-1
160 IF (S=4 OR S=5 OR S=6)
      AND Y<212-R1 THEN Y=Y+1
170 IF (S=2 OR S=3 OR S=4)
      AND X<256-R1 THEN X=X+1
180 IF (S=6 OR S=7 OR S=8)

```

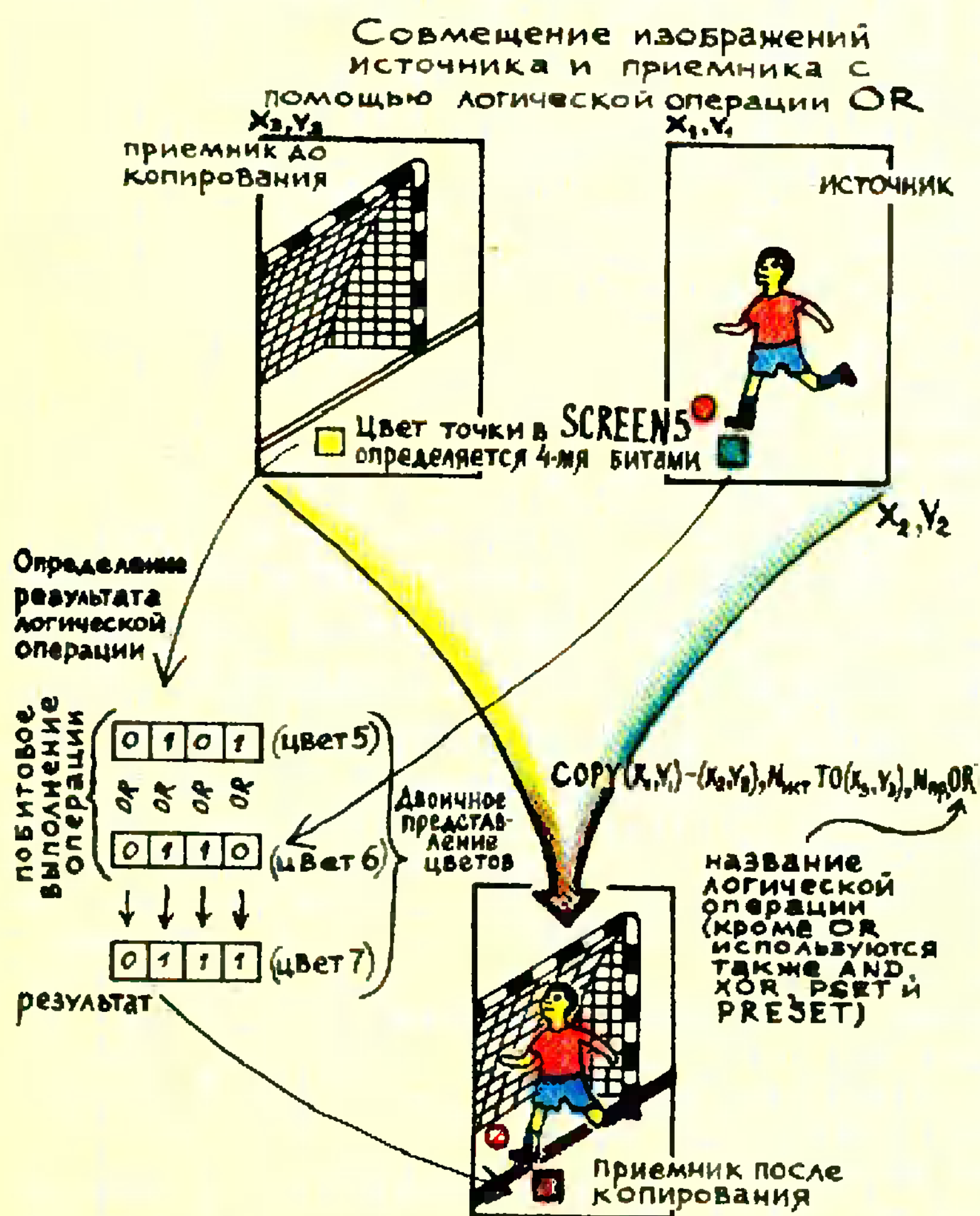
```

AND X>R1 THEN X=X-1
190 COPY (0,0)-STEP(2XR1,2XR1),2
      TO (X-R1,Y-R1),K1
200 SWAP K0,K1
210 GOTO 120

```

Движение объектов по фону

Двухэкранный способ организации движения объектов позволяет естественным образом осуществить движение объектов по некоторому фону «без его порчи». Для этой цели фон строится на третьей странице. При переключении отображаемой и активной страниц вместо оператора CLS следует выполнить копирование фона на активную страницу. Если объект рисуется, то никаких дополнительных проблем не возникает. Если же объект копируется на активную страницу, то команда COPY может «запортить» фон. Именно на этот случай при использовании COPY предусмотрен вариант совмещения изображений с помощью логических операций (рис. 6).



Пример 12. Движение большой закрашенной окружности на фоне решетки.

```

10 COLOR 10,1,1:SCREEN 5
20 FOR I=0 TO 3
30 SET PAGE I,I
40 CLS
50 NEXT I
60 SET PAGE 2,2

```

```

70 FOR X=0 TO 240 STEP 8
80 FOR Y=0 TO 200 STEP 8
90 LINE(X,Y)-STEP(8,8),4,B
100 NEXT Y,X
110 SET PAGE 3,3
120 CIRCLE(64,64),64,2
130 PAINT(64,64),2
140 N1=1:N2=0
150 FOR X=64 TO 191
160 SET PAGE N1,N2
170 COPY (0,0)-(255,211),2 TO (0,0),N2
180 COPY (0,0)-(128,128),3
      TO (X-64,48),N2,OR
190 SWAP N1,N2
200 NEXT X
210 P$=INPUT$(1):SCREEN 0

```

В ряде случаев условия задачи предполагают изменение фона по мере движения объектов по экрану. В этом случае изменения должны производиться на той странице, где хранится фон.

Пример 13. Построение циклоиды с предварительным вычислением координат точек циклоиды.

```

10 CLS:LOCATE 8,15:PRINT"ВЫЧИСЛЕНИЕ
      КООРДИНАТ"
20 N1=0:N2=1:DIM XT(255),YT(255)
30 FOR X=0 TO 255
40 R=25
50 XT(X)=X-RXSIN(X/R)
60 YT(X)=100+RXCOS(X/R)
70 NEXT X
80 COLOR 10,1,1:SCREEN 5
90 SET PAGE 2,2:CLS
100 FOR X=0 TO 255
110 SET PAGE N1,2
120 PSET(XT(X),YT(X)),7
130 SET PAGE N1,N2:CLS
140 LINE (0,101+R)-STEP(255,0),2
150 COPY (0,0)-(XT(X),150),2
      TO (0,0),N2,OR
160 CIRCLE(X,100),R
170 LINE(X,100)-(XT(X),YT(X)),6
180 SWAP N1,N2
190 NEXT X
200 P$=INPUT$(1):SCREEN 0

```

Пример 14. При построении астроида на вычисления тратится большее время. Подобно примеру 7, желательно произвести вычисления один раз и записать результаты на диск. Первая программа создает файлы «XO», «YO», «XASROID» и «YASTROID». Отметим, что использование команды COPY в этом случае не связано с предварительно построенным изображением на экранной странице.

```

10 DIM X(629),Y(629),XO(629),
      YO(629):R=25
20 CLS:LOCATE 8,10
30 PRINT "ВЫЧИСЛЕНИЕ КООРДИНАТ"
40 FOR U=0 TO 629
50 XO(U)=128+(100-R)XCOS(U/100)
60 YO(U)=105-(100-R)XSIN(U/100)
70 X(U)=XO(U)+RXCOS(U/R-U/100)
80 Y(U)=YO(U)+RXSIN(U/R-U/100)

```

```

90 NEXT U
100 COPY XO TO "XO"
110 COPY YO TO "YO"
120 COPY X TO "XASTROID"
130 COPY Y TO "YASTROID"
140 END

```

Вторая программа использует результаты работы предыдущей, что ускоряет ее работу.

```

10 COLOR 10,1,1:SCREEN 5:N1=0:N2=1:R=25
20 FOR I=0 TO 2
30 SET PAGE 0,I:CLS
40 NEXT I
50 CIRCLE(128,105),100
60 DIM X(629),Y(629),XO(629),YO(629)
70 COPY "XASTROID" TO X
80 COPY "YASTROID" TO Y
90 COPY "XO" TO XO
100 COPY "YO" TO YO
110 FOR U=0 TO 629
120 SET PAGE N1,2
130 PSET(X(U),Y(U)),7
140 COPY(0,0)-(255,211),2 TO (0,0),N2
150 SET PAGE N1,N2
160 CIRCLE(XO(U),YO(U)),R
170 LINE(XO(U),YO(U))-(X(U),Y(U)),6
180 SWAP N1,N2
190 NEXT U
200 GOTO 200

```

Быстрое построение фона и создание движения изменением фона

Использование команды COPY в ряде случаев позволяет значительно сократить время построения регулярного фона.

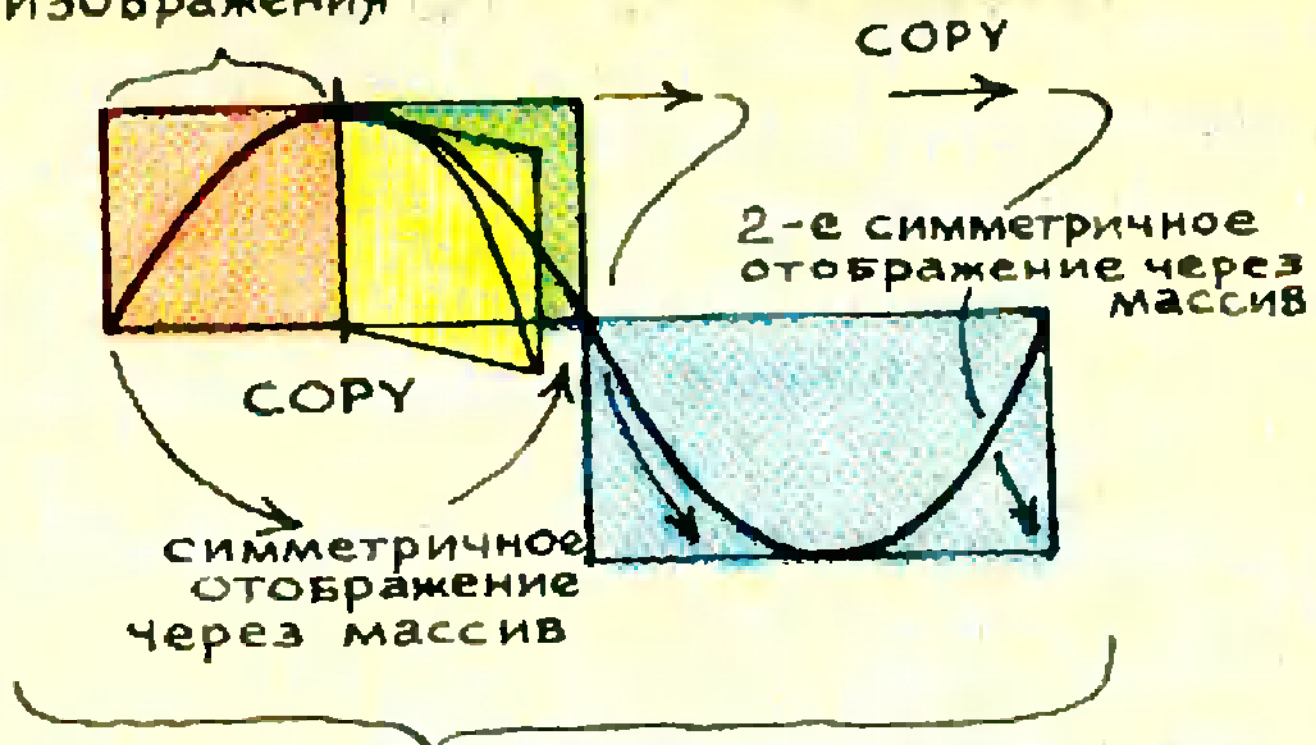
Пример 15. Построение фона, образованного синусоидами (рис. 7).

```

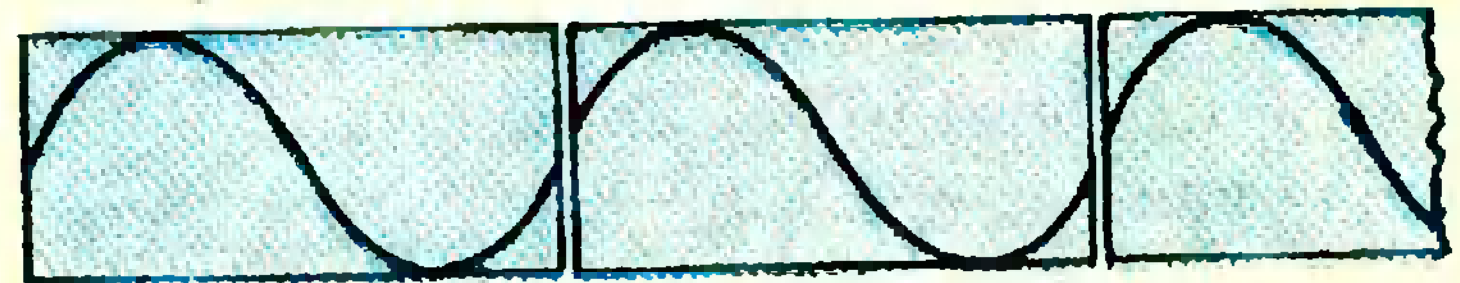
10 COLOR 15,1,1:SCREEN 5:PI=4*ATN(1)
20 R=INT((4*(30+1)*(32+1)+7)/8)+4
30 DIM SN(R)
40 FOR I=0 TO 3
50 SET PAGE I,I
60 CLS
70 NEXT I
80 FOR X=0 TO 16 STEP .25
90 Y=30-30*SIN(X*PI/32)
100 PSET(X,Y)
110 NEXT X
120 COPY(0,0)-STEP(16,30),3 TO SN
130 COPY SN,1 TO (31,0),3
140 COPY(0,0)-STEP(32,30),3 TO SN
150 FOR X=0 TO 64*3 STEP 64
160 COPY SN,0 TO (X,0),3
170 COPY SN,3 TO (X+63,60),3
180 NEXT X
190 SET PAGE 0,0
200 FOR Y=-50 TO 200 STEP 10
210 YP=Y:YH=0:YL=60
220 IF Y<0 THEN YP=0:YL=60:YH=-Y
230 IF Y>150 THEN YL=210-Y
240 COPY(0,YH)-(255,YL),3
TO (0,YP),0,OR
250 NEXT Y
260 P$=INPUT$(1):SCREEN 0

```

Исходный фрагмент изображения



Период синусоиды многократным копированием позволяет построить синусоиду во всю длину экрана



Одним из способов создания движения на экране является изменение фона.

Пример 16. Бегущие «помехи».

```

10 SCREEN 5
20 SET PAGE 0,3:CLS
30 LINE(4,4)-(12,12),,BF
40 FOR X=0 TO 16*15 STEP 16
50 FOR Y=0 TO 16*11 STEP 16
60 COPY(0,0)-(15,15),3 TO (X,Y),0
70 NEXT Y,X
80 SET PAGE 3,3:CLS
90 SL=RND(-TIME)
100 FOR Y=4 TO 16*11+4 STEP 16
110 DX=INT(7*RND(1))-3
120 DY=INT(7*RND(1))-3
130 C=INT(14*RND(1))+2
140 LINE(20+DX,Y+DY)-STEP(8,8),C,BF
150 LINE(4,Y)-STEP(8,8),,BF
160 NEXT Y
170 SET PAGE 0,0
180 FOR X=0 TO 16*15 STEP 16
190 X1=X-16:IF X1<0 THEN X1=16*15
200 COPY(0,0)-(15,211),3 TO (X1,0),0
210 COPY(16,0)-(31,211),3 TO (X,0),0
220 FOR T=1 TO 50:NEXT T
230 NEXT X
240 GOTO 180

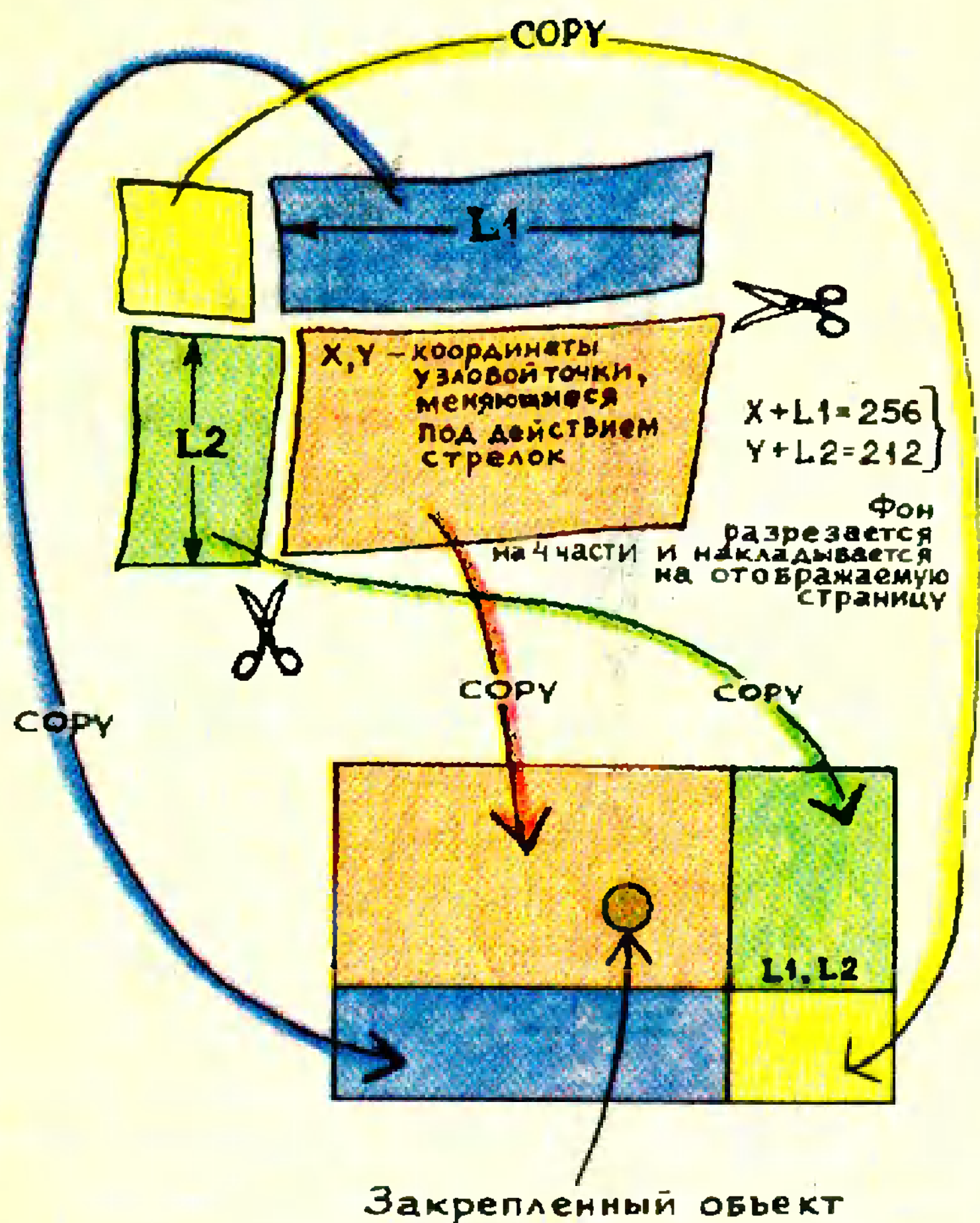
```

Наконец, важным способом получения иллюзии движения объекта может служить изменение фона при закреплённом объекте.

Пример 17. Управление движением объекта производится клавишами-стрелками. В соответствии с тем, какая стрелка или комбинация стрелок нажата, объект меняет свои очертания. Шаблоны

объектов хранятся на странице 2. Фон хранится на странице 3. При копировании разрезается на 4 куска и налагается на активную страницу (рис. 8).

Движение фона



```

10 COLOR 7,1,1:SCREEN 5
20 FOR I=0 TO 3
30 SET PAGE I,I:CLS
40 NEXT I

```

```

50 FOR R=8 TO 160 STEP 8
60 CIRCLE(128,106),R
70 NEXT R
80 SET PAGE 2,2
90 FOR I=0 TO 7
100 X=I*32+16:U=I*3.14159/4
110 CIRCLE(X,16),5,15
120 PAINT(X,16),15
130 X1=X+16*SIN(U)
140 Y1=16-16*COS(U)
150 LINE(X,16)-(X1,Y1),15
160 NEXT I
170 X=0:Y=0:K0=0:K1=1
180 SET PAGE 0,0
190 COPY (0,0)-(255,211),3 TO (0,0),0
200 COPY (0,0)-(32,32),2
    TO (128-16,106-16),0,OR
210 SET PAGE K0,K0
220 S=STICK(0)
230 IF S=0 THEN 220
240 IF S=8 OR S=1 OR S=2 THEN Y=Y-8
250 IF S=2 OR S=3 OR S=4 THEN X=X+8
260 IF S=4 OR S=5 OR S=6 THEN Y=Y+8
270 IF S=6 OR S=7 OR S=8 THEN X=X-8
280 IF X>=256 THEN X=X-256
290 IF X<0 THEN X=X+256
300 IF Y>=212 THEN Y=Y-212
310 IF Y<0 THEN Y=Y+212
320 XF=(S-1)*32
330 L1=256-X:L2=212-Y
340 COPY (X,Y)-STEP(L1,L2),3
    TO (0,0),K1
350 COPY (X,0)-STEP(L1,Y),3
    TO (0,L2),K1
360 COPY (0,Y)-STEP(X,L2),3
    TO (L1,0),K1
370 COPY (0,0)-STEP(X,Y),3
    TO (L1,L2),K1
380 COPY (XF,0)-STEP(31,31),2
    TO (128-16,106-16),K1,OR
390 SWAP K0,K1
400 GOTO 210

```

В. БУЦИК

Обучение младших школьников началам информатики

Программа «Таблица»

Она состоит из серии программных модулей, позволяющих реализовать достаточно емкий класс задач на освоение логических структур формирования линейных таблиц, поиска элементов и сортировки. Каждому модулю соответствует дисплейный кадр (см. распечатки).

Результатом деятельности школьников по решению предлагаемых задач в диалоге с компьютером должно стать уяснение таких принципов обработки, как повторение действий (поочередный

просмотр элементов таблицы, фиксирование нужных элементов, обмен значениями элементов в таблице ...) и альтернативный выбор действий (при сравнении элементов, при распознавании элементов ...).

<=				=>	
2	4	6			
1	2	3	4	5	6 7

О К Н О

ЗАДАНИЯ: ПРОДОЛЖИ РЯДЫ

- ▶ 2, 4, 6, ...
- 3, 5, 7, ...
- 5, 9, 13, ...
- 12, 15, 18, ...
- 4, 8, 12, ...
- 3, 7, 11, ...
- 1, 4, 7, ...

Окончание. Начало см.: Информатика и образование. 1991. № 2.

Вместе с тем такая деятельность позволяет развивать наблюдательность, формировать такие навыки, как поиск связей и различий в данных, вывод закономерностей и др.

Программа «Кроссворд»

Эта программа предполагает формирование базы терминов и их пояснений для кроссворда; компьютерное формирование таблицы кроссворда и выбор слов; распечатку кроссворда; разгадывание кроссворда в диалоге с компьютером.

Разгадывание кроссвордов есть одно из увлекательных и широко распространенных занятий. Не менее интересными являются и процессы их создания. Именно эти два аспекта в отношении к кроссвордам положены в основу нашего обучения.

С точки зрения информатики кроссворды привлекают тем, что предполагают создание специальной базы данных (списка терминов и их пояснений), позволяют развивать конструктивное мышление и тем самым естественным образом включать школьников в процессы знакомства с новыми информационными технологиями.

Общий вид программного меню:

ВЫБЕРИТЕ РЕЖИМ	
▶ ЧТЕНИЕ	СЛОВ
ЗАГРУЗКА	СЛОВ
ДОЗАГРУЗКА	СЛОВ
СЧИТЫВАНИЕ	СЛОВ
ФОРМИРОВАНИЕ	ПОЛЯ
ВЫБОР	СЛОВ
ПЕЧАТЬ	КРОССВОРДА
РЕШЕНИЕ	КРОССВОРДА

Покажем возможности использования программы для решения упражнений различных типов.

Первый тип упражнений (формирование базы терминов и их пояснений для кроссворда) является по сути первым опытом формирования базы данных для достаточно сложной системы. Необходимо как можно более широко представить области, в которых могли бы проявлять свои «творческие» силы школьники. Это могут быть изучаемые предметы (математика, чтение, природоведение...), области интересов (музыка, танцы, спорт...), общезначимые области (типа: Человек, Земля, Вселенная). Не-

<=	11	38	59	68	96	32	45	=>
	1	2	3	4	5	6	7	

О К Н О

46 ЧИСЛО

НЕТ ОТВЕТ

ПОИСК ЧИСЛА В ТАБЛИЦЕ
необходимо просмотреть таблицу и при обнаружении в окне такого же числа нажать пробел

<=	11	38	59	68	96	39	44	=>
	1	2	3	4	5	6	7	

О К Н О

ЧИСЛА

38 2

68 4

ПОИСК ЧИСЕЛ С ЗАДАНЫМИ СВОЙСТВАМИ:
ЧЕТНЫЕ ЧИСЛА

необходимо просмотреть таблицу и при обнаружении в окне такого числа нажать пробел

<=	10	48	57	68	12	13	55	=>
	1	2	3	4	5	6	7	

О К Н О

СЧЕТЧИК

1

ПОДСЧЕТ ЧИСЕЛ С ЗАДАНЫМИ СВОЙСТВАМИ:
НЕЧЕТНЫЕ ЧИСЛА

необходимо просмотреть таблицу и при обнаружении в окне такого числа нажать пробел

<=	62	54	25	14	10	71	54	=>
	1	2	3	4	5	6	7	

О К Н О

МЕНЬШЕЕ

14 4

БОЛЬШЕЕ

1 62

ПОИСК
наибольшего и наименьшего значений

необходимо просмотреть таблицу и при обнаружении в окне такого числа нажать одну из клавиш

<=	35	48	57	18	12	13	55	=>
	1	2	3	4	5	6	7	

О К Н О

35 1<

СОРТИРОВКА ЗАПОМИНАНИЕМ ЭЛЕМЕНТА;
УПОРЯДОЧЕНИЕ ТАБЛИЦЫ ПО УБЫВАНИЮ

<=	11	38	59	68	96	32	45	=>
	1	2	3	4	5	6	7	

О К Н А

СОРТИРОВКА ОБМЕНОМ ДВУХ ЗНАЧЕНИЙ ТАБЛИЦЫ;
УПОРЯДОЧЕНИЕ ТАБЛИЦЫ ПО ВОЗРАСТАНИЮ

Для обмена двух значений в окнах нужно нажать пробел

<=	11	38	59	68	96	32	45	=>
	1	2	3	4	5	6	7	

О К Н А

СЧЕТЧИК

1

ПОДСЧЕТ ПАР С ЗАДАНЫМИ СВОЙСТВАМИ:
ОБА ЧИСЛА В ОКНАХ ЧЕТНЫЕ

Для учета двух значений в окнах нужно нажать пробел

обычность работы заключается в том, что от учащихся требуется подбор или составление наиболее четких пояснений для выбираемых терминов. Например: «Луна — что это такое?» (и возможный ответ: «Спутник планеты Солнечной системы») или «Компьютер — что это такое?»

Организовать работу можно различным образом: фронтальная работа (учащимся предлагается термин и тут же подыскивается наиболее удачное пояснение), работа группами (каждая группа формирует свою базу данных по выбранной области или одна из групп формирует базу терминов, а другие создают базу пояснений к ним), индивидуальная работа (при всем многообразии форм ее организации).

18 Второй тип упражнений (формирование таблицы кроссворда на компьютере и выбор слов) является наиболее простым и интересным: здесь проявляется богатейшая фантазия ребят. Моднo, конечно, предлагать задания с определенными требованиями: симметрия, моделирование геометрических фигур (квадратов, трапеций ...) и т. д.

Третий тип упражнений (распечатка кроссворда на принтере) позволяет продемонстрировать возможности компьютера в получении кроссворда в виде напечатанного текста.

Четвертый тип упражнений (разгадывание кроссворда на компьютере) позволяет организовать интеллектуальную работу учащихся. Компьютерное решение имеет вариант, позволяющий отыскивать решение по отдельным буквам. Это позволяет в трудных случаях производить механический перебор вариантов.

Программа

«Графический редактор для младших школьников»

Она позволяет рисовать и конструировать рисунки на экране дисплея, сохранять полученные изображения на диске. В качестве «карандаша» могут выступать различные фигуры: точки, круги, полукруги, прямоугольники, треугольники, резинка. Редактор располагает следующими возможностями:

выбор карандаша (F1 — поиск вперед, F2 — поиск назад);

изменение цвета карандаша (F3);
изменение цвета фона (F4);
очистка экрана (F5);
запоминание рисунка (F6);
считывание рисунка (F7).

Редактор позволяет выполнять изображения по предлагаемому рисунку, а также по своему замыслу.

Как известно, современные графические редакторы для различных компьютеров с различными графическими системами обладают большими возможностями в создании и редактировании изображений: обычно предусматривается выбор графических примитивов, установление их размеров, выбор цветовой гаммы, стирание ошибочно проведенных линий и выполненных рисунков и т. д.

Вместе с тем изучение таких редакторов порой затруднено из-за освоения большого числа операционных действий. Поэтому, на наш взгляд, более целесообразно предлагать учащимся начальной школы более простые редакторы, моделирующие лишь отдельные функциональные возможности «профессиональных» редакторов. В них операционные навыки не должны затенять деятельность, связанную с построением изображений.

Предлагаемый графический редактор построен именно по этому принципу. Как показывает проведенный эксперимент, освоение младшими школьниками режимов его работы происходит за очень короткое время: 4—7 мин в I классе. Это позволяет очень быстро включать школьников в работу по выполнению различных рисунков.

Мы считаем, что для учеников в такие редакторы должны стать их «профессиональным» инструментом. Поэтому работа по созданию таких редакторов требует глубокого осмысления и должна выполняться достаточно квалифицированно (в данном случае мы в большей степени делаем упор на реализацию идей, к сожалению, порой в ущерб программному исполнению).

Данный графический редактор позволяет развивать сформулированное ранее положение об оперировании учащимися понятиями «объект», «имя», «значение». Например, можно, предлагая учащимся реальные объекты (игрушечные

автомобили, домики и т. п.), обучать их созданию (конструированию) изображения на экране дисплея, затем запоминанию изображений под некоторым именем, вызову (считыванию) изображения по указанному имени.

Кроме того, включение редактора в учебную деятельность позволит естественным образом знакомить учащихся с различными геометрическими фигурами и оперированию с ними.

Немаловажное значение имеет методика обучения работе с графическим редактором. Опираясь на имеющийся опыт, можно предложить такой вариант. Вначале учащимся рассказывается о режимах работы, предлагается испытать некоторые режимы (например, выбор карандаша, изменение его цвета, изменение цвета фона изображения), затем предлагается выполнить конкретное задание по образцу. После этого обсуждаются режимы запоминания и вызова выполненных рисунков. В дальнейшем необходимо разнообразить деятельность учащихся: предлагать для копирования различные простые и сложные рисунки, затем проводить построение рисунков реальных объектов.

Программа

«Текстовый редактор для младших школьников»

Она является простейшей моделью профессионального редактора. Позволяет учащемуся писать или вызывать небольшие тексты и редактировать их.

Редактор имеет следующие возможности:

- F1 — изменение цвета символов;
- F2 — изменение цвета фона;
- F3 — разъединение строки;
- F4 — соединение строк;
- F5 — очистка;
- F6 — копирование строк;
- F7 — поиск синтаксических ошибок;
- F8 — перенос слов и предложений;
- F9 — вызов текстов;
- F10 — запоминание текста.

Как и в предыдущем случае (при создании графического редактора), в данный текстовый редактор заложена концепция моделирования отдельных функциональных возможностей «про-

фессиональных» редакторов. Однако в нем операциональные навыки не затемяют деятельность, связанную с работой над текстами. Мы ориентируемся на создание «профессионального» инструмента для младшего школьника, т. е. стремимся обеспечивать «комфорт» в работе с компьютером, поэтому закладываем в него такие возможности как поиск и высвечивание ошибок. Это позволяет разнообразить упражнения. Например, включить написание диктантов на компьютере с последующим исправлением ошибок в тексте (при нажатии клавиши F7 ученик проверяет наличие синтаксических ошибок: компьютер автоматически отыскивает и высвечивает их цветом отличным от изображения других символов). Например, компьютер проверяет правописание слов и слогов, правила переноса. В качестве заданий могут предлагаться (аналогичные имеющимся в учебниках) наборы слов, на которых можно создавать предложения (при редактировании используются клавиши F3, F4, F6, F8 и клавиша «пробел»).

Программа

«Дневник наблюдений за погодой»

Она представляет «профессиональный» инструмент для младшего школьника для решения практической задачи: ведение компьютерной записной книжки. Позволяет формировать такие понятия, как «данные», «корректировка данных» и т. п.

Правила работы с программой довольно просты: клавишами управления курсором можно перейти на любое поле (год, месяц ...), зафиксироваться на поле можно нажатием клавиши «пробел», аналогично можно выйти из этого режима. Изменение значений (год, месяц, день, облачность ...) также производятся клавишами управления курсором.

Считаем, что создание и внедрение в обучение таких «естественно» необходимых для младшего школьника программ есть один из эффективных путей включения учащихся в изучение начал информатики.

В заключение отметим: в данной статье мы, исходя из выдвинутой об-

щей концепции обучения младших школьников началам информатики, во многом лишь наметили перспективы программных разработок, а предлагаем их примерные варианты исполнения и формируемое для них методическое обеспечение. В настоящее время нами развертывается на базе СШ № 19 и СШ № 66 г. Омска экспериментальное обучение, которое позволит в полной мере оценить достоинства как конкретных программных средств, так и выбранной концепции.

К сожалению, размеры данной статьи не позволяют ознакомить читателей с серией из десяти уроков по включению учащихся IV класса в деятельность, связанную с ознакомлением с компьютером, блок-схемами, школьным учебным языком.

20

Л. ПЛЕУХОВА

Структура и содержание мотивационного обеспечения обучающих программ

В современных условиях возрастают требования к интеллекту и мировоззрению человека, его научному и культурному уровню, профессиональному мастерству и творческим способностям; следовательно, возникает проблема: сформировать у каждого члена общества потребность в постоянном повышении своего образовательного уровня (мотивацию учения). Наличие такой потребности — необходимый признак гармонично развитой личности. Решая важную задачу — научить молодых людей учиться самостоятельно — необходимо сделать акцент на развитие индивидуальных особенностей каждого, в том числе и индивидуальных особенностей мотивации.

Массовый выпуск и поставка различных видов ЭВМ во все уровни образования, разработка и внедрение автоматизированных обучающих систем и диалоговых учебных программ создают условия для повседневной работы с ЭВМ

Автор будет благодарен за отзыв на данную работу, предлагает всем заинтересованным лицам сотрудничество.

Адрес: 644099, г. Омск. Омский педагогический институт. Кафедра информатики и вычислительной техники. В. А. Буцик.

Л и т е р а т у р а

1. Ершов А. П. Компьютеризация школы и математическое образование // Математика в школе. 1989. № 1.

2. Кузнецов А. А., Долматов В. П. Методическая система обучения ОИВТ: структура и функции, состояние и перспективы // Информатика и образование. 1989. № 1.

3. Смирнов Е. К концепции обучения информатике в младших классах // Информатика и образование. 1988. № 2.

4. Волкова С. И. Концепция курса математики для 1—4 классов // Начальная школа. 1989. № 1.

в процессе общеобразовательной и профессиональной подготовки. Однако в публикациях большое внимание уделяется главным образом прикладным проблемам компьютеризации, и сравнительно мало публикаций, посвященных анализу этой проблемы на психолого-педагогическом уровне, позволяющем раскрыть весь комплекс вопросов, связанных с организацией учебно-познавательной деятельности обучаемых: формированием мотивации и познавательного интереса в учении, установлением рационального, педагогически оправданного диалогового общения обучаемых с компьютером, сочетанием индивидуальных, групповых, коллективных форм компьютерного обучения и т. д. [1].

Необходимость учета мотивации при компьютеризированном обучении отмечается как в работах основоположников программированного обучения Б. Скиннера, Г. Паска, Б. Льюиса,

П. Ходжа, так и в современных исследованиях Н. Ф. Талызиной, О. К. Тихомирова, А. Я. Савельева, Л. В. Стриковой, Е. И. Машбица. Тем не менее в настоящее время нет таких обучающих программ, в которых бы эти рекомендации учитывались в полной мере. Это связано в значительной степени с отсутствием обоснованной методики формирования мотивации при использовании средств НИТ в обучении.

Было бы неразумно не использовать те дидактические возможности, которыми располагает ЭВМ как средство формирования мотивации: активизация самостоятельной работы обучаемых, ускорение процесса применения знаний и умений в новых ситуациях, с новыми целями, обеспечение необходимой обратной связи, возможность оперативной диагностики мотивации, объективность и обоснованность оценки, индивидуализации обучения и т. д.

Предлагаемый подход основан на идее построения мотивационного обеспечения к обучающим программам. Поскольку доля автоматизированного обучения сравнительно мала по сравнению с традиционными методами обучения, то, говоря о формировании мотивации учения, мы предполагаем формирование ситуативной мотивации, которая формируется в ходе самого акта деятельности. При рассмотрении мотивации любого действия всегда можно обнаружить как устойчивые, так и ситуативные мотивационные образования, но они участвуют в побуждении действия не как отдельные и независимые факторы, а как переходящие один в другой моменты единого процесса формирования мотивации. Мотив как непосредственный побудитель акта деятельности обладает конкретным предметным содержанием, вписан в текущую ситуацию, соизмерим по времени своего существования с протеканием данного акта. Для нашего исследования важным явилось положение об уникальности и неповторимости ситуативного мотива, его становления в ходе самого акта деятельности, о его диалектической связи с устойчивыми мотивационными образованиями. Формирование ситуативных мотивов мы приравниваем к формированию установки на конкрет-

ную деятельность и исходим из общепсихологических положений о формировании такой установки, которая реализуется в четыре этапа: создание высокой личностной эмоциональной заинтересованности, введение многочисленных мотивов, способных потенциально определить предыдущее состояние, формирование приемлемых для данного человека мотивов, реализация выработанных установок в деятельности.

Кроме того, мы рассматриваем проектирование мотивационного обеспечения к обучающим программам как многоуровневый процесс, в котором выделяем так же, как и в обучающих программах, следующие уровни проектирования: концептуальный, технологический, операциональный и уровень реализации [2]. Если для обучающих программ на концептуальном уровне задается модель обучения, представляющая из себя описание как обучающей деятельности, так и деятельности обучаемых, включающее все компоненты обучения, психологические механизмы и принципы обучения, то для мотивационного обеспечения на данном уровне задается гипотетическая модель мотивационной сферы того контингента обучаемых, для которых создается данная обучающая программа. Гипотетическая модель строится на основе теоретического анализа мотивационной сферы людей данной возрастной категории и представляет из себя:

словесное описание, определяющее каждый уровень сформированности мотивации у данного контингента обучаемых;

набор словесных характеристик, описывающих каждый уровень сформированности мотивации с точки зрения слабой выраженности одного из основных свойств мотивации (идейно-нравственной направленности, предметной направленности и динамических свойств) [3].

На технологическом уровне проектирования обучающих программ описание обучения с уровня психологических принципов, положенных в основу проекта, переводится на уровень технологии обучения, т. е. дается описание метода и условий обучения. Для мотивацион-

ного обеспечения на этом уровне проектирования необходимо задать способы уточнения гипотетической модели. Здесь дается подробное описание способа уточнения как отдельных характеристик мотивационной сферы обучаемых, так и всей модели в целом.

Операциональный уровень проектирования обучающих программ предполагает описание процесса обучения как решения дидактической задачи с подробной фиксацией всех функций обучающей деятельности, которые возлагаются на компьютер. Для мотивационного обеспечения на данном уровне проектирования составляется описание тех его блоков, которые должны быть реализованы на компьютере. Дается описание программного блока тестовой диагностики, определяющего исходные уровни мотивации с выявлением наиболее слабо развитого. Блок позволяет дифференцировать процесс формирования мотивации, так как составлен на основе уточненной гипотетической модели мотивационной сферы обучаемых. Проводится подробное описание остальных блоков, построенных в соответствии с сущностью и этапами процесса формирования мотивации учения, а также с учетом структуры мотивационной основы деятельности обучаемого. Другими словами, на данном уровне проектирования разрабатывается описание структуры и содержания мотивационного обеспечения к данной обучающей программе.

Уровень реализации проектирования обучающих программ содержит педагогическую и программную компоненты. Педагогическая реализация содержит систему обучающих воздействий, описанных в виде сценария, содержащего алгоритм перехода от одного обучающего воздействия к другому. На уровне программной реализации происходит перевод этого алгоритма на алгоритмический язык.

Для построения модели мотивационной сферы необходимо определить всю совокупность мотивов, побуждающих к учению. Учитывая, что сущность мотивации характеризуется совокупностью признаков, отражающих интегративные свойства мотивации, на основе теоретического анализа мотивационной

сферы обучаемых выделяются основные признаки, характеризующие идейно-нравственную направленность мотивации, предметную направленность и динамические свойства. В зависимости от степени выраженности этих признаков задаются уровни сформированности мотивации, указывающие, насколько развита мотивация у разных обучаемых. Кроме того, составляется набор словесных характеристик, описывающих каждый уровень с точки зрения слабой выраженности одного из интегративных свойств мотивации. Составленное таким образом полное описание мотивационной сферы обучаемых берется в качестве гипотетической модели.

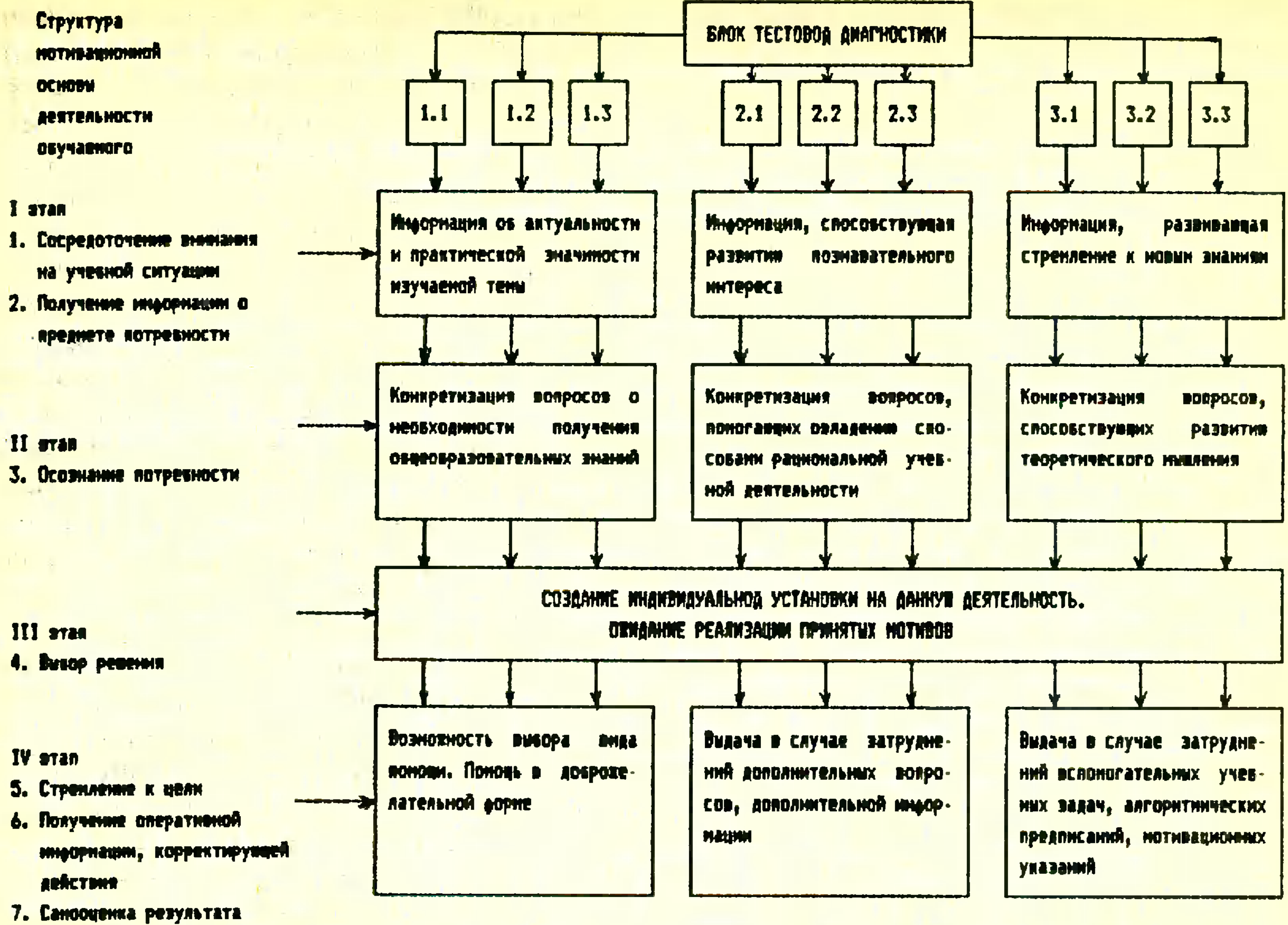
Построенная модель уточняется с помощью таких методов педагогического исследования, как шкалирование, наблюдение, обобщение независимых характеристик и самооценка обучаемых.

Опробованная и уточненная модель мотивационной сферы используется для построения мотивационного обеспечения к обучающей программе.

Мотивационное обеспечение имеет блочную структуру (см. рис.).

Первым важным блоком, с которого начинается режим обучения, является блок тестовой диагностики. Выходными параметрами блока тестовой диагностики являются номер уровня сформированности мотивации и номер наиболее слабого свойства мотивации, которые используются для выдачи на экран текста, соответствующего результатам тестирования, а также для формирования номеров меток программных блоков мотивационного обеспечения.

Программные блоки мотивационного обеспечения строятся в соответствии с уровнями мотивации, с учетом каждого свойства мотивации и согласно общепсихологическим принципам поэтапного формирования установки на учебную деятельность. Функциональное назначение блоков мотивационного обеспечения состоит в том, чтобы создавать индивидуальное отношение к различным группам обучаемых, вызывать у них интерес к работе с обучающей программой и поддерживать этот интерес на протяжении всей работы с программой за счет учета индивидуальных



особенностей их мотивации. Для реализации этой цели в каждый блок мотивационного обеспечения включен свой набор информационных кадров, вспомогательных обучающих воздействий, кадров комментариев.

Информационные кадры в мотивационном обеспечении используются главным образом на I этапе формирования установки на деятельность, связанном с созданием высокой личностной эмоциональной заинтересованности обучаемых в предстоящей деятельности. На II этапе обучаемым выдается информация, обеспечивающая введение мотивов, потенциально способных определить существующие мотивационные образования. Результатом первых двух этапов формирования установки на деятельность является III этап, который означает создание индивидуальной установки на данную деятельность. На этом этапе обучаемый ожидает реализации принятых мотивов. Задачей III и IV этапов, на котором, собственно, и происходит реализация выработанной установки, является поддержание этой установки через индивидуальные вспомогательные воздействия, на которые обу-

чаемый уже настроен. В блоках мотивационного обеспечения для этих этапов должен содержаться набор различных для каждого уровня мотивации вспомогательных воздействий по всем контрольным вопросам, выдаваемым обучаемому.

Обращение к блокам мотивационного обеспечения происходит только в том случае, если обучаемый дал неверный ответ на вопрос. Когда разные группы обучаемых по несколько раз отвечают неправильно на один и тот же вопрос, то они получают различные пояснения к своему ответу с различной степенью подсказки.

В заключение отметим, что в эксплуатируемых в настоящее время обучающих программах не существует специальных средств для формирования мотивации учения у обучаемых. Наличие таких средств повысит эффективность автоматизированного обучения.

Литература

1. Гершунский Б. С. Компьютеризация в сфере образования: проблемы и перспективы. М.: Педагогика, 1987.

2. Машиц Е. И. Психологические основы управления учебной деятельностью. Киев: Вища школа, 1987.

Вниманию руководителей учебных заведений!

Если вы заинтересованы, чтобы ваш заведующий кабинетом учебной вычислительной техники получал доплату за обслуживание оборудования, приобретайте

«Сонет-02»!

В настоящее время учебно-производственная ассоциация средних специальных учебных заведений «Учтехпром» совместно с предприятиями, учебными заведениями начала производство компьютерных классов «Сонет-02», предназначенных для оснащения средних специальных учебных заведений, вузов, общеобразовательных школ, профессионально-технических училищ.

Этот класс представляет собой эффективные аппаратные и программные средства для обучения информатике, общеобразовательным и специальным предметам.

Краткая характеристика:

- ▲ экран 25 строк по 64 или 80 символов;
- ▲ графика: 192×128 точек, три градации яркости;
- ▲ накопители на гибких магнитных дисках общей емкостью 1,6 М байт;
- ▲ локальная сеть: конфигурация «Шина», скорость обмена 19,2 К байт/с;
- ▲ удобное и дружественное сетевое программное обеспечение, не имеющее аналогов среди других компьютерных классов;
- ▲ дисковая операционная система, совместимая с CP/M, на всех компьютерах в классе;
- ▲ инструментальные программные комплексы для создания обучающих курсов;
- ▲ профессиональные программные средства для ведения баз данных, экономических и бухгалтерских расчетов, редактирования текстов.

Рабочее место преподавателя включает в себя модуль вычислительный центральный (МВЦ), накопитель на ГМД, принтер «Электроника М 6313», профессиональный монохромный монитор «Электроника МС 6105».

Рабочее место учащегося состоит из МВЦ и видеомонитора «Электроника МС6105».

По заказу системы программирования могут быть на языках ассемблер, Паскаль, Си, Фортран, Бейсик.

Компьютерный класс «Сонет-02» — это рабочее место преподавателя и от 4 до 30 рабочих мест учащихся, объединенных в локальную сеть.

Ассоциация «Учтехпром» берет на себя монтаж класса под ключ, обучение преподавателей, сопровождение программного обеспечения, гарантийное в течение 12 месяцев и послегарантийное обслуживание, консультации специалистов.

Поставки оборудования и монтаж класса производятся по государственным расценкам (класс на 16 позиций стоит приблизительно 51 тыс. рублей). Поставка гарантируется в течение месяца со дня заключения договора. Ремонт производится заменой неисправной техники на новую в течение 15 дней со дня получения заявки.

Для членов ассоциаций «Специалист», «Учтехпром», общеобразовательных школ руководство ассоциации «Учтехпром» оплачивает в течение всего гарантийного срока заведующему кабинетом 3 (три) рубля за каждый работающий компьютер и 10 (десять) рублей за рабочее место преподавателя.

Преподаватели-пользователи «Сонета-02» могут по контракту с ассоциацией «Учтехпром» разрабатывать учебные программы по своим предметам с соответствующей оплатой.

При дальнейшем совершенствовании компьютерного класса модернизация будет производиться «Сонет-02» на льготных условиях.

Гарантийные заявки следует направлять по адресу: Москва, п/о 121609, а/я 11, НПО «Коминсервис».

М. АЛЕКСЕЕВ

Пролог-Д на «Агате»

Пролог-Д — учебная версия языка логического программирования Пролог, разработанная и реализованная на «Ямахе» и БК-0010 в Йошкар-Оле С. Г. Григорьевым с сотрудниками. Изучение основ Пролога-Д является важной частью одного из новых школьных курсов ОИВТ [1]. К сожалению, до сих пор очень многие представляют себе программирование состоящим в основном из Бейсика или, в лучшем случае, из процедурных языков типа Паскаля. Логическое же программирование открывает для учащихся, да и для преподавателей, как бы новое измерение в информатике. Кроме того, знакомство с Прологом, как с языком построения баз знаний и запросов к ним, предопределяет гораздо более осмысленное обращение с традиционными базами данных и экспертными системами.

Нам очень хотелось показать Пролог своим ученикам, но найти какие-либо сведения о его реализациях на «Агате» не удалось. Пришлось реализовать самим. За помощь в этом автор признателен С. Г. Григорьеву, В. В. Малахову, А. Г. Уртминцеву и В. А. Цикозе.

Описание Пролога-Д, приемы программирования и элементы методики преподавания изложены ранее в серии статей [2—5]. В этом Агат-версия практически ничем не отличается от ямаховской, поэтому ниже будут описаны в основном особенности пользовательского интерфейса и системной реализации Пролог-Д на «Агате».

Особенности языка Пролог-Д на «Агате»

Имена фактов, правил, переменных могут иметь неограниченную длину. Числа могут быть не только целыми, но и дробными, возможна экспоненциальная форма представления. Конструкция «список» пока не реализована. Не допускается использование одноименных предикатов с разным числом аргументов. Запись выражения (факта, правила, вопроса) завершается символом «;».

В строке может быть несколько таких выражений. Допускается структурная запись правил, пробелы и символы перевода строки между предикатами игнорируются. Имеются следующие встроенные предикаты:

БОЛЬШЕ (А, В), МЕНЬШЕ (А, В), РАВНО (А, В), НЕ (предикат), ! (отсечение), ИСТИНА, ЛОЖЬ, ПС (перевод строки), ВЫВОД (переменная/число/текст), ВВОДСИМ (текст), ВВОДЧИС (число), СЛОЖЕНИЕ (А, В, С), УМНОЖЕНИЕ (А, В, С), ТОЧКА (Х, У, ЦВЕТ'), ЗАКРАСКА (Х, У, Ц), ЛИНИЯ (Х, У, U, V, Ц), ПРЯМОУГОЛЬНИК (Х, У, U, V, Ц), ОКРУЖНОСТЬ (Х, У, R, Ц), для управления системой ОТВЕТЫ (на экран/на бумагу), ОПТИМИЗАЦИЯ (нет/есть), ТРАССИРОВКА (нет/на экран/на бумагу), ЭКРАН (режим) (аналог процедуры РЖИ Рапиры).

25

По умолчанию для графических предикатов включается режим SGK — цветная графика: X от 0 до 127, Y от 48 до 127, совмещенная с шестью текстовыми строками. Переключение графической и текстовой страниц производится правой функциональной клавишей «8».

Загрузка системы

Система «Пролог-Д» работает на ПЭВМ «Агат» с дисководом на 140К или 840К в Рапира-интерпретаторе системы «Школьница» версии 1.3 (дата версии 17.10.86 или более поздняя). После загрузки Рапира-интерпретатора и получения приглашения «#» следует ввести команду

ЗАПУСК ПРОЛОГ;

Система «Пролог-Д» занимает почти всю память ПЭВМ, поэтому, если до ее запуска вы работали с другими Рапира-программами, вам следует перевызвать систему командой

РАПИРА;

и затем уже запускать Пролог.

Во время загрузки на экране дисплея последовательно появляются две красочные

заставки. Последняя предупреждает о том, что для повторного входа в Пролог-Д из Рапира-интерпретатора (приглашение «*») следует вводить
ПРО();

Если же произойдет сбойный выброс (из-за нехватки памяти) и в верхней строке появится слово ОШИБКА, для возврата следует вводить
выход;ПРО();

При входе в Пролог-Д в верхней строке экрана появляется строка основного меню
Диалог Редактор База Режим Выход
по которому клавишами-стрелками ВЛЕВО, ВПРАВО можно перемещать инверсный прямоугольник-указатель и делать выбор клавишей ВВОД.

«Выход» — это выход из «Пролога-Д» в Рапира-интерпретатор.

Диалог с системой

26 Выбрав в основном меню «Диалог» и получив в нижней строке экрана приглашение «:», можно вводить факты, правила и вопросы, разделенные и завершающиеся символом «;». При вводе можно, как и в Рапира-интерпретаторе, перемещать курсор по экрану клавишами-стрелками, используя ранее набранные строки, а также применять клавиши РЕД (включение/выключение «читающего» курсора), F2 и F3 (удаление и вставка). Ошибочные утверждения и вопросы система отвергает, синтаксически правильные утверждения (факты и правила) запоминает в базе знаний, на вопросы пытается отвечать. Например:

```
: МАМА (ЛИЗА, НАТАША);
: МАМА (НАТАША);
МАМА с 1 аргументом
должно быть 2
: МАМА (НАТАША, ТАНЯ);
: ?МАМА (ЛИЗА, ТАНЯ);
нет
: ?МАМА (ЛИЗА, Д);
Д=НАТАША
: ?МАМА (М, Д);
Ошибка в МАМА (М, Д)
: ?МАМА (М, Д);
М=ЛИЗА
Д=НАТАША
М=НАТАША
Д=ТАНЯ
: ?МАМА (_, ДОЧЬ');
ДОЧЬ=НАТАША
ДОЧЬ=ТАНЯ
: ?ПАПА (ДОМ, 19);
не знаю
: БАБУШКА (Б, В) <- МАМА (Б, М), МАМА (М, В);
: ?БАБУШКА (К, ТАНЯ);
К=ЛИЗА
: СЛОЖЕНИЕ (6, 10, 16);
Смените имя,
СЛОЖЕНИЕ - встроенный предикат
: ?СЛОЖЕНИЕ (6, 10, 16);
да
```

```
: ?УМНОЖЕНИЕ (-4, X, 7);
X=-1.75
```

Нажатие клавиши ВВОД без утверждения или вопроса возвращает систему в основное меню.

Работа с текстами программ

При выборе в основном меню поля «Редактор» включается стандартный текстовый редактор системы «Школьница», настроенный на работу с Пролог-системой.

Здесь привычным для работавших с этим редактором образом действуют все функциональные клавиши и директивы меню редактора: КАТАЛОГ, СЧИТАТЬ ФАЙЛ, ЗАПИСАТЬ ФАЙЛ, СТЕРЕТЬ ФАЙЛ, ДОПИСАТЬ ФАЙЛ, РАБОТАТЬ С ДЗУ 1—2, ОЧИСТИТЬ БУФЕР, ВЫДАТЬ НА БУМАГУ, ЗАПОМНИТЬ ТЕКСТ. Отличие от редактора, вызванного из Рапиры, проявляется в двух директивах.

Директива ВЫЙТИ позволяет вернуться в основное меню Пролог-системы (а не в Рапиру) без анализа редактируемого текста.

По директиве КОНЕЦ ОПИСАНИЯ включается не Рапира-интерпретатор, а дополнительное меню Пролог-интерпретатора: следует ли очистить базу знаний перед интерпретацией новой Пролог-программы или же база знаний будет дополняться. Затем происходит интерпретация Пролог-программы с фиксацией ошибок, если таковые встречаются. Ошибочные утверждения не включаются в базу знаний. Можно отказаться от интерпретации программы, нажав в дополнительном меню клавишу РЕД. С обработкой или без обработки программы система возвращается в основное меню.

При первом входе в редактор буфер пуст, при выходах тексты программ сохраняются в буфере и могут быть использованы при последующих входах. Исключение составляют случаи использования графики, когда буфер редактора очищается.

Работа с базой знаний

При выборе в основном меню «База» появляется меню работы с базой знаний:

Сводка Удалить Очистить базу

«Сводка» — краткая сводка содержимого текущей базы знаний. Например, после вышеприведенного диалога она будет такой:

```
Факты
2 МАМА (ЛИЗА, НАТАША)
Правила
1 БАБУШКА (Б, В)
```

Видно, что в текущей базе знаний — два факта с именем МАМА, показан первый факт, и понятно, что новые факты с этим

именем должны иметь два аргумента. В базе знаний есть также правило БАБУШКА — двухаргументное и одновариантное. Если же мы в диалоге дополним определение правила БАБУШКА отцовской линией

: БАБУШКА (Б, В) <- МАМА (Б, П), ПАПА (П, В);

это правило станет двухвариантным, и в «Сводке» перед ним появится 2.

После нажатия любой клавиши в «Сводке» выдаются также сведения о встроенных предикатах системы.

«Удалить» — позволяет удалить все факты и правила с указанным именем.

«Очистить базу» — позволяет уничтожить все факты и правила со всеми именами, возвращая базу знаний к исходному состоянию.

Встроенные предикаты защищены от удаления и очистки.

Режимы работы системы

При выборе в основном меню «Режим» появляется меню режимов работы системы:

Оптимизация Трассировка Ответы

При выборе одного из режимов ниже появляется меню его возможных состояний, текущее состояние выделено инверсно.

«Ответы» — «на экран/на бумагу» — режим вывода ответов системы на вопросы. Обычно ответы выводят на экран, но бывает нужен протокол или ответ не помещается на экране. Естественно, что для вывода на бумагу принтер должен быть включен.

«Трассировка» — «нет/на экран/на бумагу/» — во время поиска в базе знаний ответа на вопрос на бумагу могут выводиться промежуточные запросы и ответы на них. Глубина вложенности запросов отображается отступами с точками. Например:

```

: ФАКТОРИАЛ (Ø, 1) <- !;
: ФАКТОРИАЛ (N, F) <- СЛОЖЕНИЕ (M, 1, N),
:                               ФАКТОРИАЛ (M, E),
:                               УМНОЖЕНИЕ (N, E, F);
: ?ФАКТОРИАЛ (3, Ø);
. ?СЛОЖЕНИЕ (M, 1, 3)
. M= 2
. ?ФАКТОРИАЛ ( 2, E)
.. ?СЛОЖЕНИЕ (M, 1, 2)
.. M= 1
.. ?ФАКТОРИАЛ ( 1, E)
... ?СЛОЖЕНИЕ (M, 1, 1)
... M= Ø
... ?ФАКТОРИАЛ ( Ø, E)
... E= 1
... ?УМНОЖЕНИЕ ( 1, 1, Ø)
... Ø= 1
.. E= 1
.. ?УМНОЖЕНИЕ ( 1, 2, Ø)
.. Ø= 2
. E= 2
. ?УМНОЖЕНИЕ ( 2, 3, Ø)
. Ø= 6
Ø= 6

```

Сообщения трассировки выводятся на экран зеленым цветом, встроенного предиката ВЫВОД — желтым, а диалог обычно идет в голубом цвете, если это не изменено управляющими символами или функциональными клавишами. Режим трассировки очень удобен не только при отладке программ, но и при объяснении учащимся принципов поиска решений.

«Оптимизация» — «нет/есть» — режим запрещения/разрешения изменения порядка рассмотрения составляющих правило утверждений. Такие перестановки могут отрицательно повлиять на результаты рассмотрения некоторых графических и вычислительных правил, однако обычно рассмотрение с оптимизацией идет гораздо быстрее, так как при этом в первую очередь рассматриваются наиболее определенные утверждения, дерево поиска невелико и до менее определенных утверждений очередь либо не доходит (при отрицательном результате очередного рассмотрения), либо они становятся более определенными благодаря результатам предыдущих рассмотрений. Например, без оптимизации из двух вопросов

?МАМА (Б, М), МАМА (М, ТАНЯ);
?МАМА (М, ТАНЯ), МАМА (Б, М);

ответ на второй вопрос будет получен быстрее, чем на первый. Включив трассировку, увидим:

```

: ?МАМА (Б, М), МАМА (М, ТАНЯ);
. ?МАМА (Б, М)
. Б=НАТАША
. М=ТАНЯ
. Б=ЛИЗА
. М=НАТАША
. ?МАМА (ТАНЯ, ТАНЯ)
. нет
. ?МАМА (НАТАША, ТАНЯ)
. да
Б=ЛИЗА
М=НАТАША

```

В этом случае система сначала собирает все имеющиеся пары Б и М, т. е. все, что связано именем МАМА (таких пар может оказаться много, это займет время и память), потом перебирает в этих парах вторые элементы и проверяет их уже на первом месте в паре с элементом ТАНЯ, находит все-таки М — НАТАША и с ним Б — ЛИЗА. Более сложные примеры без оптимизации могут вообще не дать результата — не хватит памяти.

```

: ?МАМА (М, ТАНЯ), МАМА (Б, М);
. ?МАМА (М, ТАНЯ)
. М=НАТАША
. ?МАМА (Б, НАТАША)
. Б=ЛИЗА
М=НАТАША
Б=ЛИЗА

```

Здесь сразу однозначно определено значение М — НАТАША, а затем по нему — значение Б — ЛИЗА.

С оптимизацией же первый вопрос будет обрабатываться в том же порядке, что и второй:

```
: ?МАМА (Б, М) , МАМА (М, ТАНЯ) ;
. ?МАМА (М, ТАНЯ)
. М=НАТАША
. ?МАМА (Б, НАТАША)
. Б=ЛИЗА
М=НАТАША
Б=ЛИЗА
```

Впрочем, на оптимизирующие перестановки тоже уходит какое-то время.

Рассмотрим еще правило:

```
: ПЕЧАТЬ (СТРОКА') <-ВЫВОД (СТРОКА') , ПС;
```

Без оптимизации:

```
: ?ТРАССИРОВКА (НЕТ) ;
: ?ОПТИМИЗАЦИЯ (НЕТ) ;
: ?ПЕЧАТЬ (ПРИМЕР) ;
ПРИМЕР
да
```

Здесь — перевод строки после вывода; а с оптимизацией

```
: ?ОПТИМИЗАЦИЯ (ЕСТЬ) ;
: ?ПЕЧАТЬ (ПРИМЕР) ;
```

ПРИМЕР да

— перевод строки перед выводом.

Особенности реализации на «Агате»

Интересно, что одна из первых реализаций Пролога на ПЭВМ была осуществлена в начале 70-х гг. в Марселе на предке «Агата» — «Apple-II» — на Паскале.

Описываемая версия Пролога-Д сделана на Рапире. Пролог-система использует и ДОС, и системный стек, и ввод-вывод «Школьницы», и редактор текстов, и графику Рапиры, и арифметику с плавающей запятой. Примерно 20К занимают написанные на Рапире программы диалоговой оболочки Пролог-системы и обработки Пролог-программ, баз знаний и запросов, основанные на широком использовании такого динамичного типа данных, как кортежи. В отличие от традиционного для Пролог-систем подхода, основанного на поиске решения «в глубину» с возвратами (бектрекинг), здесь применен поиск «в ширину», в один проход, когда составляющие правило предикаты анализируются с использованием всего множества значений переменных, давших согласую-

щиеся результаты при анализе предыдущих предикатов. При этом для разрешения тупиковых ситуаций или в интересах оптимизации предусмотрены изменения в порядке просмотра предикатов. Система позволяет работать с Пролог-программами объемом до 1,5—2К (50—100 предложений) и, несмотря на двойную интерпретацию, показывает удовлетворительное быстродействие. Пролог-система стала частью «Школьницы», и освоение Пролога-Д облегчается привычным для работавших с Рапирой окружением и управлением.

Возможность реализации такой системы в рамках «Школьницы» в начале работы над проектом была неочевидна, и результат в очередной раз демонстрирует мощь Рапиры.

Конечно, это далеко не инструментальный Пролог, позволяющий создавать интеллектуальные экспертные системы, но эта система обеспечивает полную поддержку машинного курса по учебнику [1]. Это подтверждает опыт работы с ней на уроках ОИВТ в VIII—XI классах нашей школы в 1990/91 учебном году.

С вопросами и заявками на приобретение системы можно обратиться по адресу: 456321, Челябинская обл., г. Миасс, пр. Октября, 25, школа № 18, Алексееву Михаилу Николаевичу.

Л и т е р а т у р а

1. Каймин В. А., Щеголев А. Г., Ерохина Е. А., Федюшин Д. П. Основы информатики и вычислительной техники. М.: Просвещение, 1989.
2. Григорьев С., Морозов М. Давайте попробуем Пролог // Информатика и образование. 1987. № 4.
3. Каймин В., Григорьев С., Угринович Н., Щеголев А. Элементы логики и начала языка Пролог // Информатика и образование. 1989. № 4.
4. Каймин В., Григорьев С. Пролог в школьной информатике // Информатика и образование. 1990. № 2.
5. Григорьев С. Программирование на Прологе-Д // Информатика и образование. 1990. № 5.

Несколько слов о «Роботроне-1715»

Сейчас, когда наш рынок постепенно наращивает предложение 16-разрядных «персоналок» (медленнее, чем того требует спрос), 8-разрядный «Роботрон-1715» прочно обосновался на рабочих столах многих управленцев, инженеров да и в школьных классах. Кроме вездесущего дефицита тому причиной хорошие эксплуатационные характеристики: простая до минимума, но качественная клавиатура, многофункциональный принтер, не утомляющий глаз дисплей. Да и по скорости обработки информации он не на последнем месте. А если речь идет о программах, основное время работы которых уходит на печать результатов, то 1715 не уступает по скорости выполнения некоторым 16-разрядным машинам (например, ТАР-34 МХ, оснащенный принтером ТМТ-120).

Имея некоторый опыт работы на «Роботроне-1715», я хотел бы дать несколько советов юным программистам. Впрочем, они могут пригодиться и не только юным.

На эту мысль меня натолкнул случай. Одна программистка показывала заказчику свою программу, которая была насыщена световыми эффектами — миганием, инверсией. Когда взмокший заказчик выразил недовольство по поводу неприятных «сполохов» на экране во время смены окон, программистка безапелляционно заявила, что это-де какие-то мелоды в системе.

Конечно же, старушка SCPX (CP/M 2.2) ни в чем не виновата. Всею виной незнание элементарных основ схемотехники ЭВМ, хотя не стоит слишком обвинять людей в том, что они не читают техническую документацию, которой нет в комплекте. Но каждый программист должен видеть хоть немного дальше клавиатуры и набора команд языка программирования.

Чтобы понять природу «сполохов» на экране, следует в общих чертах разобраться, как происходит вывод символов на экран.

Функции отображения символов возложены на специальную микросхему — контроллер CRT. Электронные схемы развертки непрерывно смещают луч по экрану кинескопа слева направо и сверху вниз, т. е. луч движется по экрану так же, как мы читаем книгу или текст с того же экрана. Когда кончается кадр («страница»), луч мгновенно возвращается в левый верхний угол экрана, и всё повторяется сначала. Таким образом, если луч не выключать, экран будет постоянно светиться. Яркостью луча управляет CRT. Его конечной целью является включение луча точно в том месте, где

находится элемент отображаемого символа (символ отображается не сразу целиком, а за несколько строк (линий) развертки) и выключение луча на «темном» фоне. Работа CRT и схем развертки строго синхронизирована во времени.

Информацию о том, какой символ следует отображать на данном знакоместе, CRT получает из специального буфера ОЗУ дисплея, адреса которого находятся в диапазоне 0F800H—0FF7FH (и далее — при использовании 25-й строки). Длина буфера $24 \times 80 = 1920$ байт. Вот эту-то область памяти, начиная с младших адресов, CRT постоянно сканирует. Есть и исключения; например, когда у микропроцессора активен сигнал WAIT, то CRT полностью занят регенерацией ОЗУ.

Когда транзитная программа печатает символ, она обычно вызывает соответствующую системную функцию. Один из специальных модулей BIOS помещает код символа по текущему адресу буфера, если символ отображаемый. Иначе могут выполняться другие операции. Например, код 0AH BIOS не помещает в буфер, а увеличивает текущий адрес буфера на 50H и корректирует положение курсора, подготавливая таким образом переход на следующую строку. А код 1BH (ESCAPE) BIOS запоминает для того, чтобы по следующим за ним двум символам сформировать либо байт кода управляющего символа, либо команду для CRT.

Каждый код CRT расшифровывает при помощи ПЗУ знакогенератора, получая непосредственную информацию о том, как должен выглядеть данный символ на экране, заодно проверяя, не является ли он управляющим. Если для этого кода не специфицирован никакой символ, выводится точка в середине предполагаемого символа. Все эти манипуляции должны совершаться очень быстро, чтобы человеческий глаз не успевал заметить движение луча.

Действительно, все происходит довольно быстро. За одну секунду дисплей отображает несколько более 50 кадров. Далеко не всякая программа успевает завершить вывод на экран за время, меньшее длительности одного кадра, тем более за время прохода одной строки.

Допустим, что вы хотите получить инверсное изображение заголовка меню. В качестве примера возьмем программу на Бейсике:

```
0 *XXX ПРИМЕР 1 XXX
10 *XXX ИНВЕРСИЯ XXX
20 TIME=200
```

```

30 INVX=CHR(27)+"^U"; ЗНАК ИНВЕРСИИ
40 ABOX=CHR(27)+"^@"; ЗНАК ОТМЕНЫ
45 '
50 PRINT CHR(12); TAB(30); INVX;
      "М Е Н Ю:";
60 FOR I=1 TO TIME: NEXT: ЗАДЕРЖКА
70 PRINT ABOX; ОТМЕНА
75 '
80 PRINT "ЗАДЕРЖКА="; TIME
90 INPUT "НОВАЯ ЗАДЕРЖКА"; TIME
100 GOTO 50: ПРОГРАММА ЗАЦИКЛЕНА,
      ПРЕРВАТЬ - CTRL/C

```

При выполнении этой программы весь экран будет засвечиваться на время, обусловленное искусственной задержкой в строке 60. Уменьшая время задержки вплоть до удаления из программы строки 60, мы сможем уменьшить время засветки. Мы даже можем поместить переменную ABOX в конец строки 50, что еще более ускорит процесс вывода на экран, но вспышки не исчезнут совсем.

30 Думаю, уже ясно, что причина плохой работы программы — в хорошей работе CRT, т. е. наша программа не успевает разместить на экране знак отмены прежде, чем луч дошел до этой позиции.

Выход очень прост. Сначала следует разместить на экране знаки отмены в тех местах, где они должны были бы располагаться при обычном выводе. Модифицируем нашу программу согласно этому условию:

```

0 *XXX ПРИМЕР 2 XXX
10 *XXX ИНВЕРСИЯ XXX
20 DEF FNC(X,Y)=CHR(27)+
      CHR(127+X)+CHR(127+Y)
30 INVX=CHR(27)+"^U"; ИНВЕРСИЯ
40 ABOX=CHR(27)+"^@"; ОТМЕНА
45 '
50 PRINT CHR(12); FNC(1,40); ABOX;
      ЗНАК ОТМЕНЫ В ПЕРВОЙ СТРОКЕ,
      СОРОКОВОЙ КОЛОНКЕ
60 PRINT FNC(1,31); INVX; "М Е Н Ю:"
70 '
80 INPUT K
90 GOTO 60: ПРОГРАММА ЗАЦИКЛЕНА,
      ПРЕРВАТЬ - CTRL/C

```

Если у вас уже имеется очень сложное меню, содержащее много управляющих знаков, и вы не желаете возиться с определением всех точек заранее, можно пойти другим путем:

```

0 *XXX ПРИМЕР 3 XXX
10 *XXX ГАШЕНИЕ XXX
20 DEF FNC(X,Y)=CHR(27)+
      CHR(127+X)+CHR(127+Y)
30 INVX=CHR(27)+"^U"; ИНВЕРСИЯ
40 ABOX=CHR(27)+"^@"; ОТМЕНА
50 DARKX=CHR(27)+"^2"; ГАШЕНИЕ
60 '
70 PRINT CHR(12); DARKX; НЕ ОТОБРАЖАТЬ
      БУФЕР
80 PRINT CHR(131); ВЫКЛЮЧИТЬ КУРСОР
90 '
100 FOR I=1 TO 15: ЦИКЛ ДЛЯ НАГЛЯДНОСТИ
110 PRINT INVX; "М Е Н Ю:"; ABOX; ЭТО
      ПОКА НЕ ВИДНО
120 NEXT
130 '

```

```

140 PRINT FNC(1,1); " "; ОТОБРАЖАТЬ
      БУФЕР
150 PRINT FNC(20,1); ВЫХОД НА
      СВОБОДНУЮ СТРОКУ
160 PRINT CHR(130); ВКЛЮЧИТЬ КУРСОР
170 INPUT K
180 GOTO 70: ПРОГРАММА ЗАЦИКЛЕНА,
      ПРЕРВАТЬ - CTRL/C

```

В переменной DARKX определен символ гашения. Начиная с места, где он напечатан (в примере — левый верхний угол экрана), символы не отображаются на экране, хотя остаются в буфере. Попробуйте для сравнения убрать строку 50.

Недостаток приведенного примера в том, что необходимо в конце вывода позиционировать курсор «в уголок». В нашем примере после этого курсор выходит на нужную строку при выполнении строки 150, но может быть так, что требуемое положение курсора неизвестно (например, при организации «окон»). Тогда можно выйти из положения другим способом, изменив строку 140:

```

140 POKE &HF000,32: ПО АДРЕСУ
      OFB00H БЫЛ КОД ГАШЕНИЯ

```

Но следует иметь в виду, что в этом случае программа становится непереносимой на «Роботрон-1715М», так как там совершенно иная архитектура памяти.

Однако существует способ убить всех трех зайцев:

```

10 *XXX ПРИМЕР 4, ГАШЕНИЕ XXX
20 DEF FNC(X,Y)=CHR(27)+
      CHR(127+X)+CHR(127+Y)
30 INVX=CHR(27)+"^U"; ИНВЕРСИЯ
40 ABOX=CHR(27)+"^@"; ОТМЕНА
50 '
60 PRINT CHR(12)
70 OUT &H19,0: ОСТАНОВ ДИСПЛЕЯ
80 '
90 FOR I=1 TO 15: ЦИКЛ ДЛЯ НАГЛЯДНОСТИ
100 PRINT INVX; "М Е Н Ю:"; ABOX; ЭТО
      ПОКА НЕ ВИДНО
110 NEXT
120 '
130 OUT &H19,&H20: ПУСК ДИСПЛЕЯ,
      ТЕПЕРЬ ВИДНО
140 '
150 INPUT K
160 GOTO 60: ПРОГРАММА ЗАЦИКЛЕНА,
      ПРЕРВАТЬ CTRL/C

```

Эта программа пойдет вообще на любой машине, где установлен контроллер дисплея 8275. Возможно, придется изменить только адрес порта.

Когда в вашей программе используются многооконные меню или экранное редактирование с переходом по полям, возникает естественная потребность в клавишах управления курсором и функциональных клавишах. Но на 1715 эти клавиши могут быть не запрограммированы в системе, и тогда

приходится строго оговаривать для пользователя либо обязательную начальную загрузку с вашего системного диска, либо запуск специальной программы, устанавливающей необходимые клавиши. Оба способа явно несовершенны. Кому же не хочется освободиться от зависимости и разных условностей? Если бы ваша программа была живой, она потребовала бы от вас того же самого, но вы можете не дожидаться всеобщей забастовки программных продуктов против узурпатора-программиста. Дайте своей программе самостоятельность!

Здесь приводится пример на Паскале, дающий представление обо всей информации, выдаваемой клавиатурой.

(*XS-*)

```
PROGRAM KEYBOARD;
  (* ПРИМЕР 5 *)
```

```
VAR STAT, KEY1, KEY2: BYTE;
    CHECK: REAL;
    AVAIL: BOOLEAN;
    CTRL, SHIFT, SISO, LOCK:
      STRING [4]; (* СПЕЦ. КЛАВИШИ *)
```

```
PROCEDURE REQUEST;
  BEGIN (* НЕПОСРЕДСТВЕННЫЙ ОПРОС
        КЛАВИАТУРЫ *)
```

```
    AVAIL:=FALSE;
    PORT [X0E]:=X10;
    STAT:=PORT [X0E];
    IF (STAT AND 1)=1 THEN
```

```
      BEGIN (* ПРИНЯТ СИМВОЛ *)
        AVAIL:=TRUE;
        PORT [X0E]:=1;
        STAT:=PORT [X0E];
        IF (STAT AND 32)=32 THEN
```

```
          BEGIN (* СБРОС ОШИБКИ *)
            AVAIL:=FALSE;
            STAT:=PORT [X0C];
            PORT [X0E]:=X30;
          END;
```

```
        END;
      END; (* REQUEST *)
```

```
PROCEDURE ANALYSIS;
  BEGIN (* АНАЛИЗ КОДА КЛАВИШИ
        И СОСТОЯНИЯ КЛАВИАТУРЫ *)
    KEY2:=PORT [X0C];
```

```
(* СОСТОЯНИЕ СПЕЦ. КЛАВИШ: *)
    IF (KEY1 AND 1)=1 THEN CTRL:='ВКЛ'
      ELSE CTRL:='ВЫКЛ';
    IF (KEY1 AND 2)=2 THEN SHIFT:='ВКЛ'
      ELSE SHIFT:='ВЫКЛ';
    IF (KEY1 AND 4)=4 THEN SISO:='ВКЛ'
      ELSE SISO:='ВЫКЛ';
    IF (KEY1 AND 8)=8 THEN LOCK:='ВКЛ'
      ELSE LOCK:='ВЫКЛ';
```

```
(* ПЕЧАТЬ РЕЗУЛЬТАТОВ ОПРОСА: *)
    CLRSCR;
    WRITELN ('СОСТОЯНИЕ СПЕЦ. КЛАВИШ: ');
    WRITELN ('CTRL=', CTRL, 'SHIFT=', SHIFT,
            'SISO=', SISO, 'LOCK=', LOCK);
    WRITELN ('КОД НАЖАТОЙ КЛАВИШИ - ',
            KEY2);
  END; (* ANALYSIS *)
```

```
PROCEDURE WHATKEY;
  BEGIN
    REQUEST; (* ПРОВЕРИТЬ СТАТУС
              КЛАВИАТУРЫ *)
    IF AVAIL THEN (* НАЖАТА КЛАВИША? *)
      BEGIN (* ДА *)
        KEY1:=PORT [X0C];
        REPEAT REQUEST; UNTIL AVAIL;
        ANALYSIS;
      END; (* НАЖАТА КЛАВИША *)
    END; (* WHATKEY *)
```

```
(* === СТАРТ ПРОГРАММЫ === *)
  BEGIN
    CHECK:=0;
    WRITE (#12, 'НАЖМИТЕ КЛАВИШУ');
    REPEAT (* ПОВТОР ДЕМОНСТРАЦИИ,
           ПОКА НЕ НАЖАТА <ESC> *)
      WHATKEY; (* ОПРОС КЛАВИАТУРЫ
               И АНАЛИЗ НАЖАТОЙ КЛАВИШИ *)
```

```
(*
  ЗДЕСЬ МОЖЕТ БЫТЬ ПРОГРАММА,
  ВЫПОЛНЯЮЩАЯ НЕКОТОРЫЕ ВЫЧИСЛЕНИЯ
  МЕЖДУ ДВУМЯ ОПРОСАМИ КЛАВИАТУРЫ
  *)
  (* ДЛЯ ПРИМЕРА ИНДИЦИРУЕТСЯ
    ПОСТОЯННО НАРАЩИВАЕМАЯ ВЕЛИЧИНА *)
  CHECK:=CHECK+1; GOTOXY (65,1);
  WRITE (CHECK:12:0);
```

```
UNTIL KEY2=27; (* НАЖАТА <ESC>? *)
  END.
```

Программа опрашивает клавиатуру и, если было нажатие, анализирует состояния специальных клавиш CTRL, SHIFT, SI/SO, LOCK, считывает код любой нажатой клавиши и выводит всю информацию на экран. Затем инкрементируется и индицируется переменная CHECK для имитации хода некоторой программы.

Эта программа может быть значительно сокращена, поскольку здесь для наглядности она разбита на процедуры и в ней использованы дополнительные переменные. Но ее алгоритм уже оптимизирован, поэтому отдельные куски из нее могут не функционировать самостоятельно. Кроме того, помните, что во время ее выполнения нельзя пользоваться стандартными процедурами ввода-вывода, так как она сама работает с портами клавиатуры.

В данном примере также имелась бы конфликтная ситуация: после вывода на экран переменной CHECK порт клавиатуры «опустошен» и маловероятно, что клавиша будет нажата до начала нашего опроса. Чтобы избежать этого, для транслятора Паскаля задана опция (*XS-*), отменяющая опрос клавиатуры во время вывода. Для интерпретатора этот алгоритм неосуществим по той же причине. При переводе на Бейсик программа аналогичного вида будет работать только после компиляции, но опять-таки нельзя в это время применять стандартный ввод-вывод, если время выполнения программы от одного опроса до следующего соизмеримо со временем вывода.

Иногда при старте программы желательно обойти некоторый ее участок (например, заставку) по личному коду, не задерживая выполнения программы (имеется в виду уже скомпилированная и загружаемая с диска программа). Здесь можно воспользоваться тем, что порт клавиатуры запоминает последний полученный байт. При запуске программы, набрав ее имя и нажав <ЕТ>, следует еще во время загрузки нажать нужную клавишу, например <ESC>. Программа на Бейсике будет выглядеть так:

```

0 *XXX ПРИМЕР 6. ВХОД ПО КОДУ XXX
10 IF INP(12)=27 THEN 40: ' ВО ВРЕМЯ
      ЗАГРУЗКИ НАЖАТА <ESC>
20 PRINT "НЕ НАЖАТА ESC"
30 END
40 PRINT "Я УЗНАЛ ВАС"
50 .....

```

Она будет работать и без компиляции. Для этого надо ее сохранить на диске, выйти в систему и загрузить Бейсик с указанием исполняемого файла:

A>BAS PRIMER6

Если требуется распознавание более длительного пароля или команды, которую ваша программа должна получить уже при старте, следует воспользоваться системным блоком FCB. Более наглядно это демонстрируется на примере Паскаль-программы.

```

PROGRAM EXAMPLE;
(* ПРИМЕР 7. ВХОД ПО КОДУ *)
VAR FCB: STRING[10] ABSOLUTE $005C;
BEGIN
  MEM[$5C]:=8;
  IF FCB='PASSWORD' THEN WRITELN('Я
      УЗНАЛ ВАС');
END.

```

При ее выполнении в строке команд системы следует набрать

A>EXAMPLE PASSWORD

и нажать <ЕТ>. Нельзя забывать, что операционная система преобразует символы

в строке команд в большие латинские, поэтому назначенный пароль должен состоять только из таких букв, цифр и знаков. И еще: после доступа вашей программы к файлам на диске содержимое этого блока скорее всего изменится.

Часто программисты пользуются разными наборами символов при выводе на экран. Но если пользователь во время работы программы включит клавишу SI/SO, то набор символов может быть обратным задуманному и маленькие русские буквы отображаются как заглавные латинские. Этого можно избежать, если использовать непосредственное переключение знакогенератора:

```

0 *XX ПРИМЕР В (НЕ ПОДХОДИТ ДЛЯ 1715M)
10 *XXX ЗНАКОГЕНЕРАТОР XXX
20 '
30 PRINT CHR$(12); "1 - ТОЛЬКО ОСНОВНОЙ
40 PRINT "2 - ТОЛЬКО ДОПОЛНИТЕЛЬНЫЙ
50 PRINT "3 - ПО СОСТОЯНИЮ SI/SO
60 K$=INPUT$(1)
70 IF K$<"1" OR K$>"3" THEN 30
80 '
90 PRINT CHR$(12)
100 PRINT "ВВОДИТЕ ТЕКСТ В ЛАТИНСКОМ
      РЕГИСТРЕ, ПЕРЕКЛЮЧАЙТЕ SI/SO"
110 PRINT "И СМОТРИТЕ РЕЗУЛЬТАТ."
120 IF K$="1" THEN BYTE=&H3E: PRINT
      "ПЕЧАТАЕТСЯ ТОЛЬКО ОСНОВНОЙ НАБОР:
130 IF K$="2" THEN BYTE=&H7E: PRINT
      "ПЕЧАТАЕТСЯ ТОЛЬКО ДОПОЛНИТЕЛЬНЫЙ
      НАБОР:"
140 IF K$="3" THEN BYTE=0: PRINT
      "НАБОР ЗАВИСИТ ОТ SI/SO"
150 '
160 WHILE ASC(K$) XOR 13
170 K$=INPUT$(1)
180 PRINT K$;
190 IF BYTE THEN OUT &H34,BYTE
200 WEND
210 '
220 GOTO 30: ' ПРОГРАММА ЗАЦИКЛЕНА

```

Окончание следует.

С. КОМАРОВ, М. МОНИН

Про Бейсик, быстроедействие и кое-что еще

Это небольшое исследование адресовано в первую очередь владельцам БК-0010 — как начинающим, так и опытным программистам. Но если вы имеете дело с другими ПЭВМ, кое-что полезное найдется и для вас.

Долгое время БК-0010 оснащался интерпретатором Фокала, о недостатках которого неоднократно писали. Энтузиасты использо-

вали загружаемые Бейсик-85, Бейсик-ДВК, Бейсик-87 и др. Наконец в июне 1987 г. появилась модель БК-0010-01 с версией Бейсик-БК (упрощенный MSX). С тех пор не утихают споры о достоинствах и недостатках Фокала и Бейсика. Одни авторы уверяют, что интерпретатор-компилятор Бейсик-БК по быстродействию «...ставит БК-0010-01 в ряд

популярных зарубежных компьютеров» (Наука и жизнь. 1987. № 8. С. 34). Другие считают, что недостатков у Бейсика-БК больше, чем достоинств, именно поэтому «...серьезных и интересных программ на Бейсике-БК так и не создано» (Наука и жизнь. 1989. № 5. С. 127). Третьи еще более категорично высказываются в пользу Фокала (Информатика и образование. 1989. № 1. С. 50).

Кто же прав? Для объективной оценки качества интерпретаторов и компиляторов за рубежом широко применяют тесты Benchmark (PC World. 1984. Vol. 7. № 8; Информатика и образование. 1988. № 2). Тесты позволяют оценить скорость выполнения основных операторов интерпретатора: математических операций, обмена с внешними устройствами, графики и т. д. Для профессиональных ПЭВМ важнее всего скорость, точность вычислений и время доступа к памяти, хотя надо уточнить, какие задачи решает данный компьютер. Хороший домашний компьютер должен иметь быструю графику, легкий для изучения и удобный для работы язык программирования. Точность вычислений здесь чаще всего не нужна.

Скорость вычислений зависит от быстродействия процессора (т. е. самого компьютера) и эффективности работы интерпретатора или компилятора (компилятор работает намного быстрее). Быстродействие процессора определяется тактовой частотой, эффективностью и гибкостью команд (машинных кодов) и числом разрядов шины данных. Так, 16-разрядный микропроцессор K1801BM1 (БК-0010, ДВК-2, PDP-11) обрабатывает информацию в два раза большими порциями, чем, например, 8-разрядные процессоры Z80 (ZX Spectrum и др.) или 8088 (IBM PC). Сравнить наборы команд этих процессоров непросто. Такая задача выходит далеко за рамки данной статьи. Желающих отсылаем к специальной литературе (для начала можно порекомендовать: В мире персональных компьютеров. 1988. № 2. С. 67—69). Процессор «понимает» только «свои» команды. Интерпретатор или компилятор преобразует программу на Бейсике или другом языке в команды процессора. Различные Бейсики могут по-разному переводить программу в машинные коды. Одни, как плохой переводчик, тратят много лишних слов (т. е. команд), пока доберутся до сути. Другие более лаконичны. По-разному могут работать подпрограммы математических действий, например перемножения вещественных чисел (если такой процедуры нет в наборе команд процессора) или извлечения квадратного корня. Еще больше могут отличаться подпрограммы (да и сами алгоритмы) обработки строковых переменных, графики, доступа к памяти. Наконец, по-разному организуется

взаимодействие между подпрограммами, распознавание ключевых слов языка и т. д. Например, в БК-0010-01 и загружаемых «вильнюсских» Бейсиках ключевые слова хранятся в таблицах интерпретатора в полном написании, хотя могут распознаваться и при сокращенной записи. Другие Бейсик-системы (например, Basic-80 в операционной системе CP/M) хранят ключевые слова в алфавитном порядке без первой буквы (а некоторые слова — и без окончания); вспомогательная таблица содержит адреса «страниц» памяти, где хранятся ключевые слова, начинающиеся на A, B и т. д. Интерпретатор сперва распознает первую букву, а затем сразу обращается к нужной «странице». «Вильнюсские» Бейсики просматривают таблицу целиком. Есть и более глубокие различия. Все зависит от особенностей данного компьютера, неизбежных ограничений на параметры интерпретатора и искусства системных программистов — разработчиков Бейсика.

Это несколько утомительное отступление позволит лучше понять, почему одни программы могут выполняться быстрее на одном компьютере, а другие на другом. Тесты Benchmark позволяют выявить как раз эти различия. Для оценки среднего быстродействия можно воспользоваться тестом из журнала «Creative Computer» (Vol. 10, № 2, Feb. 1984). Тест содержит наиболее употребительные операторы: арифметических выражений, функций и циклов. Попутно тест проверяет точность вычислений и качество генератора псевдослучайных чисел. В упомянутой статье приведена таблица результатов для 115 зарубежных компьютеров, от супер-ЭВМ до дешевых карманных моделей. Вот этот тест и его «перевод» на Фокал.

```

10 FOR N=1 TO 100
20 A=N
30 FOR I=1 TO 10
40 A=SQR(A)
50 R=RND(1)
60 NEXT I
70 FOR I=1 TO 10
80 A=A^2
90 R=R+RND(1)
100 NEXT I
110 S=S+A
120 NEXT N
130 PRINT ABS(1010-S/5)
140 PRINT ABS(1000-R)

1.10 F N=1,99; D 2
2.10 S A=N
2.20 F I=1,9; D 2.3
2.30 S A=FSQT(A)
2.35 S R=R+FRAN()
2.40 F I=1,9; D 2.5
2.50 S A=A^2
2.55 S R=R+FRAN()
2.60 S S=S+A

```

Разумеется, обе программы можно записать короче, но именно в таком виде этот тест работал на зарубежных ЭВМ.

Как работает тест? Если из любого числа натурального ряда (кроме единицы) десять раз извлечь квадратный корень, а результат десять раз возвести в квадрат, полученное число будет несколько отличаться от исходного. Чем меньше разница, тем точнее работает интерпретатор. Точность вычислений зависит от числа цифр, с которыми происходят вычисления (т. е. разрядной сетки) и алгоритмов математических операций. Переменная S есть сумма обработанных таким образом чисел от 1 до 100. Вместе с накоплением суммы происходит накопление погрешностей. Точное значение суммы 5050. Строка 130 (или 3.10) вычисляет и выводит абсолютную величину накопленной погрешности, уменьшенную в пять раз.

Переменная R равна сумме двух тысяч псевдослучайных чисел. Математическое ожидание этой величины при равномерном распределении от 0 до 1 равно 1000. Таким образом, второй результат позволяет, хотя и довольно грубо, оценить отклонение распределения от равномерного. По мнению «Creative Computer», ориентировочные оценки таковы:

точность вычислений: 0.0000001 — отлично, 0.18 — плохо;

генератор случайных чисел: 5 — отлично, 15 — плохо.

Этот тест мы запускали на БК-0010-01 и на некоторых других ЭВМ. Результаты, взятые из цитируемого журнала, помечены звездочкой, остальные получены авторами.

Результаты выполнения теста «Creative Computer» (время в минутах, секундах и десятых долях секунды) приведены в таблице.

На какие размышления наводит полученная таблица? Если говорить о БК-0010, то по точности Бейсик-БК сопоставим с мощными профессиональными ЭВМ и значительно превосходит все персональные домашние компьютеры. Но по быстродействию он уступает даже Фокалу и большинству дешевых 8-разрядных домашних компьютеров. В первую очередь это объясняется тем, что интерпретатор-компилятор БК-0010-01 вычисляет все функции с двойной точностью. Даже если заменить все переменные в тесте на переменные одинарной точности (например, S! или R!), время выполнения теста почти не меняется, а точность падает до 0.011! В утешение владельцам БК-0010-01 отметим, что программы, в которых нет

вычисления функций, выполняются быстрее аналогичных на Фокале.

Загружаемый же Бейсик-87 при сопоставимой с другими домашними компьютерами точности значительно превосходит их в быстродействии и приближается к профессиональным ЭВМ. Бейсик-ДВК оказывается ниже всякой критики, а Фокал, как видим, не только медлителен, но и дает наибольшую во всей таблице погрешность.

Тест не позволяет выявить дефекты генератора FRAN, которые отмечали многие «бэкашники». Генератор RND в Бейсике-БК вроде бы неплох. Так, во всяком случае, следует из теста. Но если FRAN генерирует коррелированные пары чисел, то RND выдает тройки! Следующая простая программа убедит вас в этом.

```
10 CLS
20 X=256XRND(1)
30 Y=240XRND(1)
40 C=3XRND(1)
50 PSET (X,Y),C
60 GOTO 20
```

Экран покроется не случайным цветным узором, а диагональными полосами разного цвета.

Вывод неутешительный: ни Фокал, ни Бейсик-БК для бытового компьютера не годятся. О том, где можно использовать Фокал, написано немало. Бейсик-БК с его высокой точностью лучше всего применять для решения математических задач с итерационными алгоритмами: интегрирования систем дифференциальных уравнений, вычисления специальных функций и т. д. У одного из авторов есть достаточный опыт применения БК-0010-01 для подобных целей, и результаты получались очень хорошими. Для «домашнего пользователя» важнее быстродействие, а подобная точность — излишество! Если к сказанному добавить, что операторы графики в загружаемом Бейсике также исполняются в несколько раз быстрее, а объектный код не «съедает» и без того скудную память, что Бейсик-БК далеко не полностью соответствует стандарту MSX и содержит системные ошибки (см., например: Информатика и образование. 1989. № 1, 2, 3; 1990. № 2, 3), вывод ясен: версию Бейсик-Вильнюс 1986.07.24 (так она именуется официально) следует признать неудачной. Если ряд БК получит развитие, на наш взгляд, для зашивки в микросхемы следовало бы принять Бейсик-87, предварительно расширив его до 16К байт дополнительными командами графики, редактирования и работы с файлами.

Что касается тестирования других компьютеров, внимательный читатель обнаружит в приведенной таблице немало интересного. Некоторые захотят повторить экспе-

ЭВМ	Время выполнения	Точность	Генератор псевдо- случайных чисел
XCray-1	0'00,01	0.0000000014	6.1
XVAX-11/780	0'01	0.11	5.3
XWANG 2200 SVP	0'05	0.000000076	3.9
XIBM PC	0'24	0.011	6.3
XDEC Professional 330	0'25	0.014	7.7
XCommodore 64	1'53	0.001	8.9
XApple IIe	1'53	0.001	12.0
XTRS-80 Model 1	2'19	0.03387	12.0
XAtari 800	2'23	0.006875	7.0
XTI 99/4a	3'46	0.00000011	2.6
XSinclair Spectrum	4'39	0.000668	3.5
XSharp PC-1211	28'32	0.000028	-
IBM PC AT (12 МГц, 80287):			
GW-Basic 3.2	0'02,70	0.005859	7.1884
Turbo-Basic 1.0	0'01,45	0.0115234	3.8839
Quick Basic 4.0	0'01,25	0.0115356	11.6106
УКНЦ (Бейсик ВГУ)	0'12	0.067299	21.0454
ДВК-3	0'18	0.00964	0.812
ЕС 1840 (Бейсик М86)	0'22	0.011597	9.3175
Искра 1030 (Basic 85)	0'25	0.011597	9.3175
СМ-3:			
Basic RT-11	0'26	0.00964	0.812
FORTRAN RT-11	0'04,50	0.00964	1.0
— с двойной точностью	0'29	0.000000000009	1.0
Электроника-60	0'27	0.00964	0.812
ДВК-2М:			
Basic RT-11	0'27	0.00964	0.812
Бейсик-ВГУ (87.09.15)	0'11	0.06073	21.0455
Электроника-85	0'29	0.014648	7.6756
Искра-226	0'42	0.000000076	1.3E-16
Роботрон 1715	1'35	0.188	7.482
Корвет	1'37	0.03387	7.701
Агат	1'55	0.00104	12.0
Правец 8Д	2'24	0.00104	6.7834
ПК-01 Львов	2'51	0.03387	1.571
ПК Криста	2'55	0.03387	1.307
Микроша	3'01	0.03387	1.55
Электроника ДЗ-28	3'47	0.00000516	1.59
Ямаха MSX-2	3'52	0.000000206	12.4944
БК-0010:			
Бейсик-БК	2'58	0.00000000112	6.E-26
Бейсик-ДВК	3'35	0.000761	12.22
Бейсик-85/87	0'14	0.0418	7.94
Фокал	1'02	0.2463	6.3244

35

римент. Не удивляйтесь, если время выполнения теста на вашем компьютере окажется немного другим: тактовая частота у каждого конкретного экземпляра может отличаться от эталонной. И еще один нюанс. Некоторые интерпретаторы (в том числе и Бейсик-БК) при запуске автоматически изменяют началь-

ную установку RND. При этом результат тестирования генератора псевдослучайных чисел каждый раз будет иным.

Было бы интересно получить и проанализировать аналогичные результаты от владельцев других типов ПЭВМ. Приглашаем к разговору!

Гармоничная учеба

Что такое научно-техническая революция? Только лишь слова на кумачовом транспаранте? Нет, во всех развитых странах это реальность. А мы отстаем, и отдельные достижения (космические и т. п.) только подчеркивают это обстоятельство. Новые измерения техники, технологии, науки не стали повседневностью.

Зато явно выражено повсеместное снижение уровня здоровья. Среди причин этого как ухудшение экологической обстановки, проистекающее, в частности, из недостаточного использования достижений технологии, так и прогрессирующий отход условий жизни от тех, к которым *Homo sapiens* был приспособлен эволюцией. Растущие нервные нагрузки, повышение спроса на «работу головой, а не руками», ширящиеся информационные потоки...

Подготовить человека к жизни в обществе, далеко ушедшем от первобытнообщинного, призвана система образования. Однако она не претерпела принципиальных изменений со времен средневековья. Ее цель — дать знания, умения и навыки по некоторой группе предметов (от чистописания до эксплуатации ядерных реакторов), причем решается эта задача устаревшими методами, что приводит к бесконечной экстенсификации процесса обучения. Результат — недостаточные подготовленность молодежи, квалификация специалистов. Все более явной становится также печальная зависимость: больше знаний — меньше здоровья.

Последней атакой знаний на здоровье стала компьютеризация. Объективно необходимая (и в школе, и в экономике), она в то же время весьма опасна как дальнейшим ухудшением (с точки зрения физиологии *Homo sapiens*) условий труда, так и иллюзией прогресса, пока, увы, не достигнутого. Автоматизированные системы обучения могут интенсифицировать процесс обучения, но могут и нанести вред, сравнимый по масштабам с Чернобылем. На Всесоюзной конференции по АОС эта мысль автора нашла активную поддержку, в решения конференции был включен соответствующий пункт.

Образованию нужен реальный прогресс, переход в новое качество, соответствующее сегодняшним запросам общества. Оно должно стать, во-первых, более интенсивным; во-вторых, более, так сказать, физиологичным, т. е. лучше соответствовать конструкции организма, не отнимать здоровье, а давать его. Говоря короче, образование должно

стать гармоничным; должно давать и широкие знания и хорошее здоровье.

Можно ли добиться сразу и того и другого, если до сих пор не удалось решить даже одну первую задачу? Казалось бы, удвоение задач должно вызвать как минимум удвоение трудностей, но...

Вспомним советского авиаконструктора итальянского происхождения Роберто Бартини. Он остался малоизвестен широкой публике: его самолеты зачастую просто не строились. Причина? Они слишком обгоняли свое время. Например, тяжелый сверхзвуковой самолет, очень близкий к «Конкорду» и ТУ-144, Бартини спроектировал в 1955 г. Зато уж если самолет строился, он оказывался рекордным, да и не по одному параметру, а по нескольким. Такой эффект давало любимое правило Бартини «И-И». Разрешение противоречий «ИЛИ-ИЛИ» (можно увеличить или дальность, или грузоподъемность и т. п.) он искал в композиции тождества противоположностей (и дальность, и грузоподъемность) — и находил! Причем каждый раз такая находка открывала целый класс принципиально новых путей.

Мной разработана и опробована адаптивная технологическая система интенсивного обучения (АТСИО), позволяющая достичь сразу двух целей — приобретения широких и глубоких знаний при одновременном улучшении здоровья. Органичным элементом АТСИО является применение новых информационных технологий активного и интенсивного (с использованием скрытых резервов организма индивида) обучения (НИТАИО).

О скрытых резервах следует сказать особо. Общеизвестно, что свои интеллектуальные способности человек эксплуатирует в лучшем случае на 10—20 %, физические — на 20—50 %. Большая часть потенциала «сберегается» природой для экстремальных ситуаций. Попытки использовать его «в лоб», экстенсификацией процесса, в том числе учебного, приводят к истощению запасов здоровья. Сопротивление организма этому процессу, обычно не проявляющееся в сознании индивида, идущее независимо от сознания, не позволяет превзойти определенный предел интенсивности обучения. Если же действовать в согласии с природой, можно использовать скрытые резервы, добиваясь одновременно как интенсификации обучения, так и благотворного влияния на здоровье. Гармония взаимоотношений с природой поз-

воляет добиться гармонического развития.

Сложная проблема — обеспечение непрерывности образования. При переходе от одной ступени обучения к другой большой трудностью для индивида является освоение новой технологии обучения. Значительно упростит дело использование единой технологической системы от детского сада до курсов повышения квалификации и самообразования. АТСИО пригодна для этого. Кроме того, АТСИО позволяет преодолеть естественный возрастной барьер уменьшения способностей к восприятию нового, возникающий обычно уже после 30 лет.

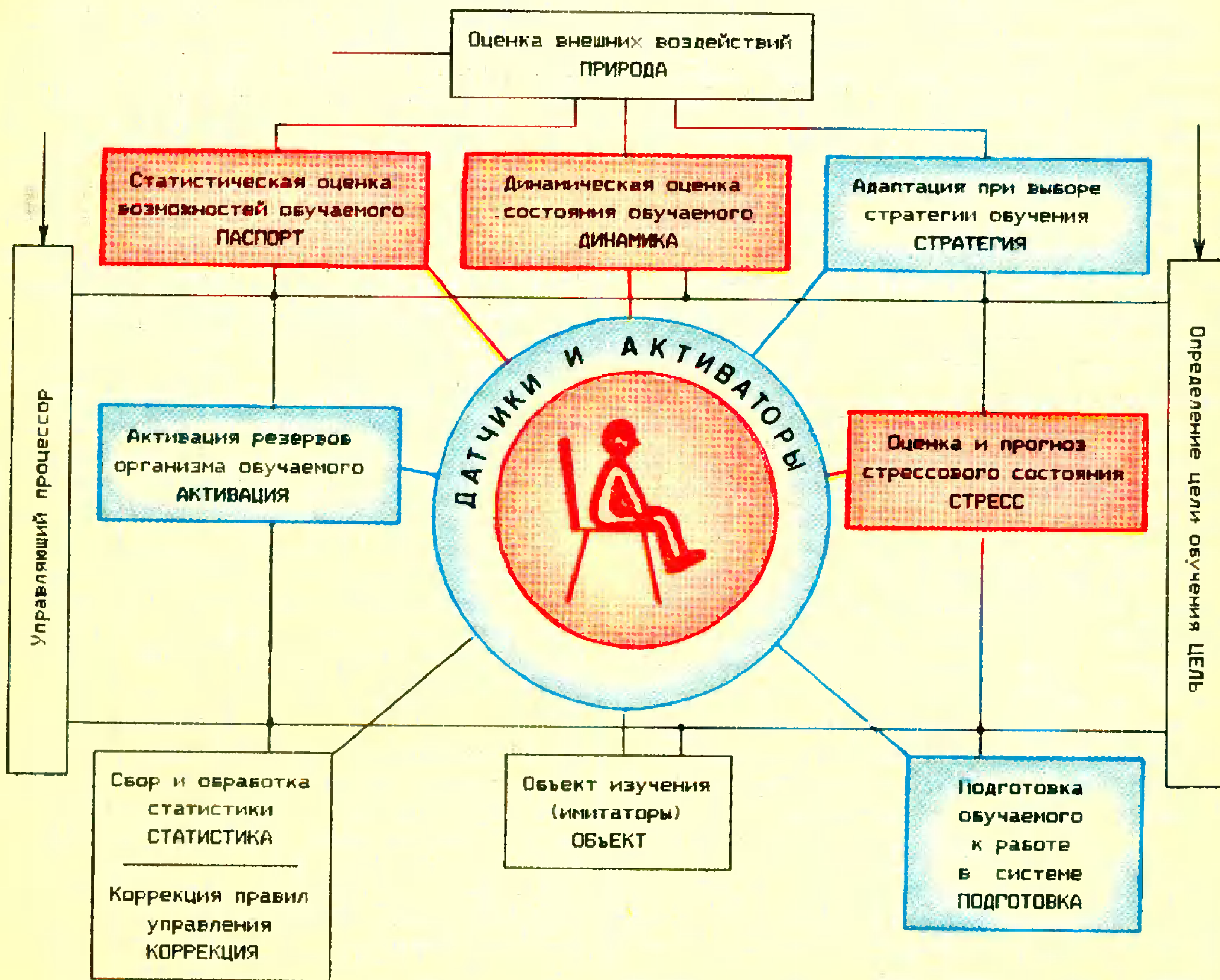
К сожалению, в рамках данной статьи описание АТСИО может быть только поверхностным. Более полная информация есть, например, на ВДНХ СССР — в филиале павильона «Народное образование» (экспонаты «Адаптивная технологическая система интенсивного обучения: основы теории построения и практика применения», «Методические указания по использованию системы интенсивного обучения начинающих пользователей ЭВМ»).

Первым действием АТСИО является «изучение» индивида, предпринимаемое еще

до начала обучения (подсистемы «Паспорт», «Природа», «Цель»). Результаты «изучения» могут использоваться для адаптации индивида в процессе подготовки к работе с АТСИО (подсистема «Подготовка») и обязательно используются в ходе интенсивной учебно-тренажерной работы. Состояние индивида постоянно отслеживается (подсистемы «Динамика», «Статистика»), система адаптирует стратегию обучения (или психофизиологического, лечебного и т. п. воздействия) с учетом факторов среды обитания, цели обучения, состояния индивида и других (подсистемы «Стратегия», «Коррекция»).

В АТСИО могут использоваться самые разные приемы воздействия, в частности, с автоматизированным или с ручным управлением (в том числе с участием индивида в выборе вариантов и воздействия, и управления).

Предположим, что система по статистическим данным определила, что тренд ошибок по вине зрительного анализатора приближается к границе допустимого. Ее реакцией может быть психофизиологическое воздействие на зрительный анализатор, снимающее усталость глаз и одновременно напряжен-



ность центральной нервной системы: прерывание учебного процесса и включение автомата, обеспечивающего выполнение специальных упражнений для снятия усталости зрительного анализатора (подобных публиковавшихся в одном из номеров «ИНФО»)*, или запуск специальной программы, обеспечивающей «игру цветов» на экране дисплея с музыкальным сопровождением, и др. Другой пример лечебного воздействия системы — распыление лекарственных аэрозолей, которые, в частности, могут быть выбраны самим обучаемым из предложенного системой списка подходящих в данном случае средств.

Для растормаживания скрытых резервов (подсистемы «Активация», «Стресс») применяются методы, средства и технологии активации (не путать с общепринятыми организационно-психологическими приемами активизации). В их число входят суггестопедические, гипнопедические, физиологические воздействия, «мозговой атаки», управления цветоцветозвуковой средой, «базовых опор» и др.

С некоторыми из них читатели «ИНФО», конечно, знакомы, о других имеет смысл кратко рассказать.

Многим приходилось наблюдать, как гипнотизер добивается от подопытного индивида проявления незаурядных, дотоле скрытых способностей. В АТСИО такой прием используется как с участием специалиста, так и с помощью технических средств (например, магнитофильма).

В суггестопедическом методе (Г. Лозанов), в отличие от гипнопедического, состояние индивидуума естественное, а информация предъявляется в суггестивной форме на субсенсорном уровне, т. е. ниже порога естественного восприятия, на уровне подсознания. Информация содержит элементы косвенного внушения, направленного на стимуляцию заданных действий индивида и на преодоление психологического барьера, препятствующего интенсивной познавательной деятельности.

В АТСИО эффективность суггестопедического метода повышается как благодаря возможности создания мощной информационной базы, ориентированной на уровень неосознаваемого уровня, так и благодаря индивидуальному подходу. Кроме того, существенно используются возможность управления психофизиологическим состоянием обучаемых с помощью суггестивных внушаю-

щих воздействий, многоплановое и многоканальное представление информации и другие приемы.

Суггестокрибернетический метод (В. Петрусинский, С. Киселев и др.) реализует суггестопедический с помощью специально разработанного авторами технического комплекса.

В АТСИО можно реализовать широкий круг как существующих, так и перспективных методов интенсивного обучения. Она предоставляет авторам новых методов неограниченные возможности для максимального и безопасного использования заторможенных природой резервов человека не только при обучении, но и при любой другой деятельности, связанной с новыми информационными технологиями.

Эффективность некоторых технологий (методов) обучения при изучении английского языка

Технология (метод)	Время (ч)	Объем (слов)	Скорость (слов/ч)
Традиционный	840	4500	5,35
Аудиолингвальный	700	3500	5,0
Аудиовизуальный	270	1500	5,55
Гипнопедический	120	1200	10,0
Погружение в языковую среду	140	1600	11,4
Релаксopedический	98	1200	12,1
Ритмопедический	96	2000	21,1
Суггестокрибернетический	80	4000	50,0

Эффективность системы интенсивного обучения (СИО) начинающих пользователей ЭВМ в сравнении с традиционным обучением (ТО)

Показатель	Значение		Соотношение СИО/ТО
	СИО	ТО	
Среднее число программ по типовым заданиям, подготовленных «средним» обучаемым за 1,5 мин	2,62	0,625	4,16
Среднее число ошибок на одну программу	0,84	3,07	0,27
Среднее число задач, решенных «средним» обучаемым с первого запуска	0,65	0,04	16,25
Доля обучаемых, не подготовивших ни одной типовой задачи	0	0,4	—
Доля обучаемых, подготовивших программы по значительно более сложному (по сравнению с типовым) заданию	0,52	0	—

* Схема автомата и алгоритм управления им от ПЭВМ, возможно, будут опубликованы в нашем журнале. — Примеч. ред.

Качество и эффективность АТСИО оказалось возможным оценить только сравнительными методами. Для этого были выбраны три дидактические системы (ДС): традиционная (показатель условно обозначен через E_0), с применением ТСО (E_1), с применением АОС (E_2), показатель АТСИО — E_3 . Для надежности использованы два способа сравнительной экспертной оценки: по 100-бальной шкале и по так называемому приоритетному списку (по каждому частному показателю строится упорядоченный ряд, кортеж, сравнительного качества конкурирующих ДС) конкурирующих ДС. На основе этих оценок получены следующие усредненные и нормированные (т. е. $E_0 + E_1 + E_2 + E_3 = 1$) показатели качества:

О. ЛУЧКО, А. ДЬЯЧЕНКО

Омский государственный педагогический институт им. А. М. Горького

Программно-аппаратная система «Светофор»

Опыт внедрения компьютера в процесс обучения дает основания считать, что использование на уроке только педагогических программных средств (ППС) в не полной мере раскрывает потенциал компьютера как универсального устройства, обеспечивающего автоматизацию различной деятельности человека. Например, при составлении и исполнении алгоритмов (программ) управления программно-реализованными роботами-исполнителями практически невозможно раскрыть учащимся все аспекты процессов, происходящих при управлении компьютерами реальными техническими устройствами. Использование в качестве исполнителей некоторых механических устройств, управляемых с помощью команд алгоритмического языка, также в не полной мере решает указанную проблему. На возможность и педагогическую целесообразность использования на занятиях по различным предметам реальных устройств, управляемых с помощью компьютера, указывают и ученые-педагоги (А. А. Кузнецов, И. В. Роберт и др.). Разработанная авторами программно-аппаратная система «Светофор» состоит из аппаратной части и пакета управляющих программ.

Как сделать «Светофор»

На рис. 1 приведена принципиальная схема устройства, используемого в комплексе с ПЭВМ «Ямаха». Устройство подклю-

$E_0=0,096$; $E_1=0,2$; $E_2=0,24$; $E_3=0,476$.

Результаты экспериментов с использованием даже неполного числа приемов активации показывают, что объем усвоенных знаний, скорость и качество обучения в АТСИО в 3—16 раз выше, чем в традиционных дидактических системах.

Мной разработаны проект автоматизированного рабочего места НИТАО и программа его оборудования и распространения, но реализация этой идеи требует создания широкой научно-технической базы, которую следует разрабатывать с участием многих научных направлений. Поэтому мы будем рады подключению энтузиастов к этой важной работе.

чается к компьютеру при помощи разъема ХР1 типа РРМ7-14Ш-ПБ-В. На физическом уровне управление им осуществляется путем подачи напряжения уровня логической единицы на заданные контакты интерфейса «Центроникс». Возможно подключение данного устройства к компьютерам других типов через интерфейс типа «IRPR».

Рассмотрим работу устройства на примере одного из каналов. С приходом уровня логической единицы на вход 1 микросхемы DD.1 на выходе 2 появится уровень логической единицы, который поступит на базу ключевого транзистора VT1 и откроет его, что приведет к срабатыванию оптрона U1 и открытию тиристора VS1 (загорится красный свет). Оptron U1 служит для гальванической развязки устройства и сети. В блоке применены резисторы МЛТ, конденсаторы К50—16. Светодиоды HL1, HL2, HL3 выведены на лицевую панель блока для визуального контроля работы каналов и снабжены соответственно подписями: «красный», «желтый», «зеленый».

Как показывает опыт, изготовление подобного устройства вполне по силам учащимся, интересующимся радиоэлектроникой.

Как управлять «Светофором»

Управление устройством «Светофор» возможно на нескольких уровнях.

Первый уровень. Рассматривая данное устройство как исполнителя, можно наде-

обеспечивающих вывод некоторых «символов» (заданного кода) на принтер.

Например:

LPRINT CHR\$ (&B00000001); — «зажечь красный, погасить зеленый и желтый»;

LPRINT CHR\$ (&B00000111); — «зажечь красный, зеленый и желтый».

Ниже приведена программа, по которой последовательно зажигаются и гаснут все лампы:

```

10 LPRINT CHR$ (&B001);
20 GOSUB 100
30 LPRINT CHR$ (&B010);
40 GOSUB 100
50 LPRINT CHR$ (&B100);
60 GOSUB 100
70 GOTO 10
100 FOR I=1 TO 100:NEXT I
110 RETURN

```

Третий уровень. Он предполагает обращение к порту ввода-вывода принтера. В данном случае от учащихся уже потребуются серьезные знания по информатике, связанные, в частности, с представлениями об архитектуре реального компьютера. Например:

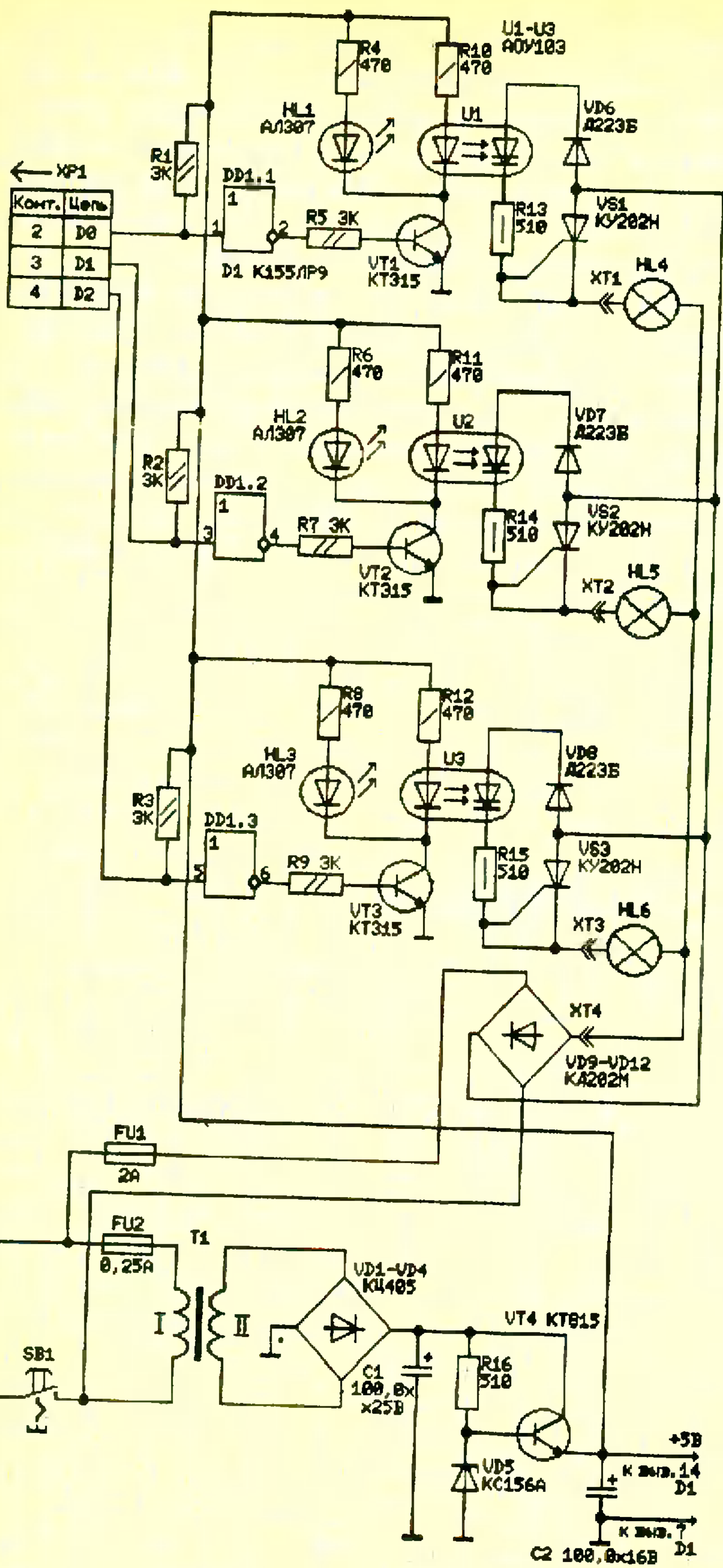
```

OUT &N91, &B001:OUT &N90, 0:OUT
&N90, 255

```

Использование при создании алгоритмов (программ) управления устройством «Светофор» описанной многоуровневой системы позволяет строить изложение учебного материала в зависимости от возраста учащихся и имеющегося уровня их подготовки в области информатики.

Основной областью использования программно-аппаратной системы «Светофор» в школе является курс информатики. В ПТУ данное устройство может использоваться при изучении курса «Автоматизация производства на основе ЭВТ». В педагогических вузах — в курсах «Методика преподавания информатики», «Использование ЭВТ в учебном процессе». Данное устройство может также использоваться в качестве управляющего при создании учебных систем, предусматривающих комплексное использование различных средств обучения: педагогических программ, аудио- и видеотехники. Так, например, обучающая программа может в заданный момент «включить» («выключить») кодоскоп или видеоманитофон. Все, кого заинтересовало данное устройство, могут связаться с авторами по телефону в г. Омске: 23-62-56.



лить его следующей системой команд (СКИ): «зажечь красный», «погасить красный», «зажечь зеленый», «погасить зеленый», «зажечь желтый», «погасить желтый». Определена также команда «задержки» на заданное время, необходимая для моделирования работы реального светофора. Авторами разработана специальная программа для создания, редактирования и исполнения соответствующих алгоритмов. Сам алгоритм представляет собой последовательность данных команд, причем предусмотрена работа учащихся как в режиме непосредственного управления (введенная команда сразу же исполняется), так и в режиме автоматического исполнения всего алгоритма.

Второй уровень. В этом случае управление осуществляется с помощью команд языка программирования MSX — Бейсик,

Учебная система управления базами данных

В учебном процессе Ухтинского индустриального института с 1982 г. используются микроЭВМ. Накоплен значительный опыт обучения студентов программированию, численным методам, опыт выполнения лабораторных работ, курсовых и дипломных проектов. Эти направления в компьютеризации учебного процесса в вузах традиционно развивались первыми, и в них использовались очевидные вычислительные достоинства ЭВМ.

В современном производстве более широко используются не вычислительные, а информационные возможности ЭВМ. Поэтому с 1986 г. в учебном процессе строительного факультета используются учебные базы данных. Были созданы СУБД «ИНПО-86» для вычислительного комплекса «Искра-1256» и СУБД «ИНП-90» для ЭВМ «Атари».

Так как обе системы носят учебный характер, то их объем ограничен оперативной памятью этих ЭВМ (64К байта). Внешняя память используется для хранения СУБД и сформированных в ходе учебного занятия баз данных.

Это позволяет проводить лабораторные работы для студентов при ограниченном количестве дисководов и даже при наличии учебного класса с одним дисководом (возможно использование базы данных с магнитофоном).

Системы «ИНПО-86» и «ИНП-90» подобны. Они состоят из ряда сервисных подсистем, обращение к которым выполняется из основного меню:

- формирование структуры;
- заполнение базы данных;
- просмотр и корректировка данных;
- выборка данных;
- сортировка по алфавиту (латинскому и русскому) и по величине;
- вертикальная обработка;

горизонтальная обработка;
горизонтальная выборка;
печать таблицы.

СУБД хранит и обрабатывает числовую и символьную информацию.

Выборка данных — это выборка хранимой информации по ряду последовательно выполняемых отношений ($<$, $>$, $=$, $<=$, $>=$, $<>$), объединенных логическими условиями типа «и» или «или».

Вертикальная обработка — вычисление суммы, среднего арифметического, среднего квадратичного и других статистических характеристик информации в графе.

Под горизонтальной обработкой понимается заполнение графы информацией, полученной обработкой по заданной формуле числового содержимого других граф, т. е. это элемент работы с электронной таблицей.

Вся работа с СУБД ведется в диалоговом режиме. Студенты осваивают работу с СУБД за 1—2 занятия, а затем выполняют индивидуальную работу по созданию небольшой базы данных и получению заданных документов обработки этой информации.

СУБД «ИНПО-86» создана на языке ВК «Искра-1256», а «ИНП-90» написана на языке Бейсик.

Это позволяет не только обучаться принципам обработки информации с помощью баз данных, но и изучать СУБД, совершенствовать СУБД и создавать новые подсистемы СУБД. Например, студенты неоднократно использовали СУБД для обработки результатов сессии и самостоятельно создавали свои версии подсистем СУБД для автоматизации получения выходных документов.

Система может быть адаптирована на другие микроЭВМ.

Адрес для справок: 169400, Коми АССР, г. Ухта, ул. Первомайская, 13, Ухтинский индустриальный институт, кафедра вычислительной техники.

В продаже — МС1502

Основа этой ПЭВМ — микропроцессор КР1810ВМ88, аналог интелевского 8088, с которым и сейчас выпускаются дешевые модели IBM PC XT. ОЗУ — 128К байт, в том числе видеоОЗУ — 32К байт. ПЗУ — 48К байт, в нем записаны Бейсик, монитор, базовая система ввода — вывода, программные средства контроля. Формат графического экрана 320×200 (4 цвета из 16 возможных) или 640×200 (один цвет из 16 возможных) точек. Внешнее запоминающее устройство — магнитофон, скорость вывода информации на МЛ — 1200 бод. Весьма высококачественное устройство генерации звуков с частотой от 50 Гц до 16 кГц. Средняя нагрузка на отказ — 10000 ч, на сбой — 500 ч, средний срок службы — не менее 10 лет. На прилагаемых МЛ записаны 16 игровых программ, текстовый редактор и база данных.

Цена компьютера (договорная) — 1400 руб., изготовитель — завод «Континент» (п. Зеленодольск Днепропетровской обл.).

М. ИВАНОВ, МАГАДАН

ДОСКА ОБЪЯВЛЕНИЙ

42 Интересуюсь компьютерами «Агат», «Вектор-06Ц»; предлагаю обмен программами. 404104, Волжский, 10 м/р, 33, кв. 119. С. Даниличев.

Хочу переписываться с владельцами ПК-01 «Львов».

283280, Тернопольская обл., г. Кременец, ул. Молодая гвардия, 8. В. Н. Толкачеву.

Предлагаю владельцам «Агатов» организовать свой клуб. Для начала присылайте идеи, предложения.

349010, Луганская обл., п. Славяносербск, ул. Ленина, 66, кв. 49. Р. И. Жага.

Желаю переписываться с владельцами ПЭВМ МС 1502.

685001, Магадан, ул. 2-я линия, 16. Иванов М. В.

Имею множество программ для БК-0010, «Атари», «Коммодор-64». Предлагаю обмен.

626603, Тюменская обл., г. Салехард, ул. Чапаева, 21, кв. 1. С. В. Ивлев.

Предлагаю обмен программами для БК. Имею около 400 программ. В первую очередь жду писем от «бэкашников» из Прибалтики и Белоруссии.

232007, Вильнюс, ул. Кривю, 29, кв. 18. П. И. Дейчко.

Ищу и предлагаю литературу по вычислительной технике и программированию для IBM PC и других ЭВМ.

398002, Липецк, ул. Семашко, 13/2, кв. 2. Е. Е. Сирош.

Предлагаю для обмена большой набор программ для «Атари ХЕ».

210027, Витебск, а/я 8. О. М. Батура.

Предлагаю обмен программами для ПК-01 «Львов», имею более 50 программ.

420095, Казань, ул. Кулахметова, 25, корп. 1, кв. 124. Д. В. Овчинников.

Ищу друга, владельца ПК «Спектр».

235804, Клайпеда, ул. Сулупес, 11а, кв. 75. Д. Л. Шилейка.

Хочу переписываться и обмениваться программами с владельцами «ZX Spectrum».

620120, Свердловск-45, ул. Строителей, 14, кв. 40. Д. Батулин.

Хотел бы обмениваться программами, опытом и информацией с владельцами компьютеров, совместимых с «ZX Spectrum».

314040, Полтава, ул. Ю. Кондратюка, 15, кв. 11. А. Белоус.

В нашей школе разработаны и успешно применяются в учебном процессе программы для машинных курсов информатики на ПЭВМ «Агат» Робот, Измеритель, Чертежник (учебник А. Г. Кушниренко и др.), Пролог-Д (учебник В. А. Каймина и др.), аналогичные «ямаховским».

456321, Челябинская обл., г. Миасс, просп. Октября, 25, СШ № 18, М. Н. Алексееву Тел. 2-14-75, код 35135.



Ю. ЗАЛЬЦМАН

Архитектура и ассемблер БК

Архитектура БК

Векторы прерываний

Мы уже не раз упоминали термин «прерывание». Пора, наконец, разобраться, что же это такое. Попробуем для начала дать общее определение. *Прерывание* — это приостанов выполнения процессором обработки текущей задачи и переход к обслуживанию внешних устройств или программных запросов. Как происходит так называемая обработка прерывания? Рассмотрим вначале обработку прерываний от внешних устройств.

Допустим, ЦП выполняет операции по обработке текущей задачи, а в это время поступает запрос на прерывание от внешнего устройства. Запрос этот обычно подается по одной из особых электрических цепей ЦП и не является командой в обычном понимании. Если приоритет ЦП, установленный для текущей задачи, меньше, чем приоритет внешнего устройства, от которого пришел запрос, происходит прерывание: процессор завершает выполнение очередной команды и прекращает ввод следующих команд, после чего записывает в стек текущее ССП (содержимое PS), а затем — содержимое РС, т. е. адрес следующей команды. Эти данные необходимо сохранить — ведь после окончания обработки прерывания процессор должен продолжить обработку текущей задачи начиная со следующей команды и с тем же самым приоритетом. Часто бывает так, что нужно сохранить не только содержимое РС и PS, а и другие данные, но тогда об этом должен позаботиться уже программист; сам процессор сохраняет только основное. Записав ССП и РС в стек, ЦП выполняет так называемый *переход по вектору прерывания*.

Вектор прерывания — это просто два машинных слова, записанных одно за другим в специально выделенной зоне ОЗУ — системной области. Первое из них представляет собой адрес начала программы обработ-

ки данного прерывания, второе — ССП, установленное для данного прерывания. Из этого описания ясно, что каждое прерывание должно иметь свой собственный вектор.

Итак, процессор, выполняя переход по вектору прерывания, переписывает его первое слово в РС, а второе — в PS, после чего, естественно, следующей выполняется команда, записанная в памяти по адресу, указанному в РС, т. е. начинается выполнение программы обработки прерывания. Приоритет ЦП при этом установлен согласно новому содержимому PS, которое, как мы помним, переписано из второго слова вектора. Если приоритет этот достаточно низкий, т. е. допускает на фоне обрабатываемого новое прерывание, то оно также может быть обработано, и т. д. Те, кто занимался практическим программированием, не могут не заметить в этом процессе некоей аналогии с обращением к подпрограмме; правда, обработка прерывания несколько осложняется изменениями ССП, а следовательно, и приоритета, и битов (флагов) условий.

Раз уж мы усмотрели аналогию с подпрограммой, то вспомним, что из любой подпрограммы должен быть и возврат к выполнению основной программы. Точно так же существует и *возврат из прерывания*, для чего в конце программы его обработки должна быть записана команда возврата из прерывания. Обнаружив ее, ЦП выполняет действия, обратные только что описанным, — восстанавливает из стека содержимое РС и PS и, естественно, продолжает выполнение прерванной программы, ведь теперь в РС — адрес следующей команды текущей задачи.

Помимо прерываний от внешних устройств существуют и так называемые *командные прерывания*, или *программные запросы*. Их обработка отличается от только что описанного процесса лишь тем, что при этом прерывание вызывается не сигналом от внешнего устройства, а особой командой, встретившейся по ходу программы. Ясно, что программный запрос на прерывание удовлет-

воряется всегда и не зависит от приоритета — не может же процессор игнорировать команду, которую сам прочитал! Командное прерывание, таким образом, еще больше похоже на переход к подпрограмме.

Векторы прерываний располагаются, как было сказано, в специально отведенной зоне ОЗУ. В БК-0010 эта зона занимает адреса 0—276. Здесь можно разместить до 48 векторов прерываний. Но такого количества векторов в БК нет, а свободные ячейки памяти используются для других целей.

Откуда же берутся в системной области ОЗУ векторы прерываний? Очень просто: они переписываются туда при запуске системы из ПЗУ. А ЦП уже «знает», по какому адресу записан каждый вектор, эти сведения заложены в него аппаратно, и при поступлении соответствующего запроса он ими пользуется.

44 Опишем кратко имеющиеся в БК-0010 векторы прерываний. «Имя» вектора прерывания обычно получает от того адреса, по которому записано его первое слово. Эти адреса строго фиксированы для данного типа процессора и потому точно определяют вектор.

Вектор 4 — прерывание по зависанию. Переход по этому вектору выполняется как по запросу внешних устройств, так и программно. Внешним устройством является клавиша **СТОП**, а программно переход по вектору 4 вызывает команда HALT — останов. Кроме того, прерывание по этому вектору может вызвать и сам ЦП при зависании (что это такое, описано выше), откуда и следует его название. Клавиша **СТОП**, в отличие от других внешних устройств, вызывает прерывание всегда (понятие приоритета на нее не распространяется).

Вектор 10 — прерывание по резервному коду. Как уже было сказано, не все возможные коды использованы в наборе команд ЦП. Если в процессоре выполнения программы ЦП встретит не известный ему код, происходит прерывание по вектору 10.

Вектор 14 — прерывание по T-разряду. Прерывание по данному вектору происходит после выполнения очередной команды, если T-разряд ССП равен 1.

Вектор 20 — прерывание по команде IOT. Используется в операционных системах.

Вектор 24 — прерывание по аварии сетевого питания. В БК данный вектор не используется, хотя программа обработки в ПЗУ имеется; просто вектор не задействован аппаратно — соответствующий вход процессора не подключен, а устройство, вызывающее прерывание при снижении напряжения питания, отсутствует.

Вектор 30 — командное прерывание EMT. Этот очень широко используемый вид

прерывания, как и следующий — по вектору 34, будет подробно описан в дальнейшем.

Вектор 34 — командное прерывание TRAP.

Вектор 60 — прерывание от клавиатуры. Запрос на прерывание по этому вектору поступает при нажатии любой клавиши, кроме **СТОП** и регистровых. Запрос удовлетворяется, если приоритет процессора менее 4. Если приоритет равен или больше 4, запрос на прерывание «зависает» до окончания обработки процессором текущей задачи или до снижения приоритета.

Вектор 100 — прерывание по таймеру. Помимо внутреннего, системного, таймера, описанного выше, к БК-001 можно подключить и внешний, или таймер по прерываниям. Его преимущество в том, что ЭВМ, работая с ним, не должна периодически опрашивать регистр счетчика, а таймер сам, например, раз в секунду вызывает прерывание. Приоритет таймера такой же, как клавиатуры, и прерывание по вектору 100 разрешено, если приоритет процессора ниже 4. Программа обработки данного прерывания предусмотрена в Фокале, содержимое счетчика таймера вызывается функцией FCLK ().

Вектор 274 — прерывание от клавиатуры по нижнему регистру. Как уже было сказано, в регистре данных клавиатуры (адрес 177662) формируется при нажатии клавиши 7-разрядный код. Его коррекция в код нижнего регистра осуществляется благодаря тому, что для обработки нажатия клавиши по нижнему регистру предусмотрен отдельный вектор прерывания. В остальном вектор 274 не имеет отличий от вектора 60.

Остальные ячейки в области векторов прерывания БК-0010 используются для других целей. Но нам вполне достаточно и имеющихся векторов. Рассмотрим несколько подробнее, что и как можно сделать с их помощью.

Командные прерывания

Мы обещали разобраться, что представляют собой прерывания по векторам 30 и 34. Настало время выполнить обещание.

Командные прерывания EMT и TRAP очень похожи друг на друга и имеют общий код: 104AAA, где число AAA — аргумент EMT или TRAP. Разница между этими командами заключается лишь в том, что аргументы EMT лежат в пределах 0—377, а TRAP — 400—777. Рассмотрим, как работают данные команды, на примере EMT.

Пусть процессор встретил в программе команду с кодом 104AAA. Первым делом он определяет, какое число «скрывается под псевдонимом» AAA, т. е. значение младших 9 битов команды. Если это число меньше 400, то команда — EMT и происходит пре-

рывание по вектору 30. Как обычно, процессор сохраняет в стеке PS и PC и переходит по адресу вектора 30. Здесь вместо конкретной, единственной, программы обработки прерывания — целый пакет программ! Но ничего ужасного не происходит — «на пороге» нас встречает маленькая, но очень важная программка: *EMT-диспетчер*.

Первым делом он выделяет младший байт команды EMT, т. е. ее аргумент. Это путеводная нить в лабиринте множества программ обработки EMT. По аргументу (или, как еще говорят, *номеру EMT*) EMT-диспетчер находит в специальной таблице адрес той единственной программы, которая нам нужна, и передает ей управление. Таким образом, задавая номер EMT, мы можем с помощью этой единственной команды обращаться к множеству программ, выполняющих самые разные функции.

Но чем такое обращение отличается от обращения к подпрограммам — ведь их тоже можно написать множество? И обращаться к каждой просто по имени метки... Вроде бы проще и понятнее, да только вот команда обращения к подпрограмме занимает в памяти два слова, а EMT — только одно! Таких обращений может быть в тексте программы много сотен, и тогда преимущество EMT — экономия памяти в сотни слов. Для БК с его, мягко говоря, не слишком емкой памятью это немаловажно.

А что же TRAP, какова роль этой команды? Практически разницы между EMT и TRAP нет, верное, разница чисто условная. Если аргумент команды 104AAA лежит в пределах 400—777, то выполняется прерывание по вектору 34. Причем в самой команде указывается, как и в EMT, номер в диапазоне 0—377, а в аргумент переводит его уже ассемблер при трансляции. Например, команде EMT 4 соответствует код 104004, а команде TRAP 4 — код 104404. Принято использовать команды EMT в системных программах ЭВМ, а команды TRAP — в пользовательских программах, только и всего. Что же касается порядка их обработки, то, как уже можно понять, он одинаков; для вектора 34 программа обработки начинается с *TRAP-диспетчера*, аналогичного EMT-диспетчеру.

Программа EMT-диспетчера БК-0010 расположена по адресу 100112 и обслуживает множество команд EMT с различными номерами. Каждая из них делает что-то свое, иногда эти действия весьма сложны и всегда полезны, причем не только ЭВМ, ими может воспользоваться и программист, тем самым избегая написания подпрограмм, длинных и очень сложных. Зачем это, если достаточно вписать в программу единственную команду EMT с нужным номером?

На что же «годны» имеющиеся в системе БК-0010 EMT, или, другими словами, что они делают?

Окончание следует.

Ассемблер БК

Псевдооператоры

Как уже неоднократно говорилось, при трансляции каждый оператор (с принадлежащими ему операндами, если они имеются) переводится в машинный код, образуя машинную команду, или инструкцию. Но имеется ряд специфических задач, для решения которых обычных операндов недостаточно, например резервирование в памяти места под константы, переменные и массивы, создание в памяти текстов, символьных и цифровых последовательностей и т. п. Для этого служат *псевдооператоры*, или *псевдокоманды*.

В отличие от обычных операторов псевдокоманды при трансляции не переводятся в машинные коды; они представляют собой лишь указания транслятору, что записать в память с текущего адреса или что выполнить при трансляции. Заносимая в память информация, если она имеется, следует после псевдооператора. Рассмотрим псевдооператоры, имеющиеся в ассемблерах МИКРО. Все они начинаются с символа «точка», который в ассемблере означает *текущий адрес*. Таким образом, любой псевдооператор можно истолковать как указание транслятору: «записать с текущего адреса...».

.E — обнуление слова или байта и приведение к четному текущему адресу. Как известно, команды ассемблера после трансляции занимают от одного до трех машинных слов, слово — это два байта, и, значит, любая команда может после трансляции начинаться лишь с четного адреса. Бывают ситуации, когда мы при написании программы не можем с определенностью сказать, четен ли текущий адрес. Это случается, например, если в текст программы вписана последовательность символов или байтовых кодов. Но чтобы перейти к дальнейшей записи операторов, необходим четный текущий адрес. Кроме того, при выводе, например, текстовых сообщений удобно отмечать конец текста нулевым байтом. Обе задачи решаются одновременно, если в конце текста (или последовательности кодов) стоит псевдооператор .E. Встречая его, транслятор заносит в память нулевой байт, если адрес нечетный, и нулевое слово, если четный. Адрес, таким образом, всегда приводится к четному. Кроме того, данный псевдооператор применяется для резервирования отдельных слов

памяти под переменные, если их исходное значение — ноль.

.+X — зарезервировать X байтов ОЗУ. Встречая в тексте этот псевдооператор, транслятор обнуляет указанное количество байтов и увеличивает текущий адрес на X байтов (X — восьмеричное число). Применяется для выделения памяти под массивы, приведения текущего адреса к нужному, «подгонки» длины программ под «круглое» число и т. п. Пусть, например, мы с помощью оператора EMT 10 вводим текст максимальной длины 20 байтов, а начало буфера текста — метка T1. Для выделения буфера под текст достаточно записать

T1: **.+20**

.#X или **.#MET** — запись константы в слово по текущему адресу. Константой может быть как восьмеричное число (без знака или со знаком «минус»), так и имя обычной метки (можно и с коррекцией адреса MET+X или MET-X). В составе псевдокоманды может быть через запятую перечислено несколько констант, например:

.#14563, TEX+140, T27, -20, 24, 0, 532

Каждая из этих констант записывается в свое слово. Если константой является адрес метки, то его абсолютное значение транслятор заносит только после компоновки программы. В менее совершенных версиях ассемблеров допускается запись после **.#** только одного значения константы. Этот псевдооператор широко применяется для записи констант, переменных (с заданием их исходного значения) и строк абсолютных адресов меток, которые можно затем использовать для адресации различных модулей программы. Приведем пример такой записи. Пусть в программе имеется 10 модулей, помеченных метками M0:, M1:, M2:, ... M9:, а директивы передачи управления этим модулям — вводимые с клавиатуры цифры 0, 1, 2, ... 9 соответственно. Как оптимально организовать блок управления такой программой? Запишем:

```
BEG: EMT 6 Ввод директивы
      ASL R0 Умножить на 2 код
;      цифры
      JMP @ADR-140(R0) Передача
;      управления
;      модулю
ADR: .#M0, M1, M2, M3, M4, M5, M6, M7, M8, M9
```

Первая строка программы вполне понятна, мы заносим с клавиатуры в R0 код директивы, нулю соответствует код 60, единице — 61 и т. д. В последней же строке — адреса меток модулей, на которые мы должны передать управление. Адрес каждой метки, естественно, занимает слово, значит, адреса

следуют с интервалом два, а коды управления 60, 61... и т. д. следуют с интервалом в единицу. Чтобы привести их в соответствие с интервалом следования меток, во второй строке мы умножаем код на 2. А вот в третьей строке, которая собственно и передает управление, использованы все возможные ухищрения адресации. Мы передаем управление по адресу, который в свою очередь, записан по адресу (двойная косвенность!), определяемому выражением $R0 + \#ADR - 140$ (140 есть не что иное, как удвоенный код нуля). Таким образом, если, например, мы вводим директиву 3, у нас получается $63 \times 2 + \#ADR - 140 = \#ADR + 6$. А по адресу $\#ADR + 6$ записан адрес метки M3, на нее и будет передано управление. Отметим, что наша программа, в принципе, еще не имеет законченного вида: если мы введем директиву, код которой больше кода девятки или меньше кода нуля, результат будет непредсказуем. Следовало бы после оператора EMT 6 проверять содержимое R0 на соответствие заданному диапазону кодов, например, так:

```
CMPB '0', R0
BNI BEG
CMPB '9', R0
BLO BEG
```

Но в первоначальном тексте программы мы этот фрагмент не привели, заботясь о максимальной простоте.

.B: — запись константы в байт по текущему адресу. В качестве констант могут использоваться только восьмеричные числа от 0 до 377. Несколько чисел после одного псевдооператора могут быть записаны через запятую (в ранних версиях — только одно число). Эта псевдокоманда применяется в основном для записи строк кодов, используя которые можно выполнить ряд действий. Можно также, записав строку кодов управления, найти в ней заданный код (директиву) и передать управление на модуль программы, пользуясь строкой меток, например так:

```
BEG: EMT 6 Ввод директивы
      MOV #DIR, R4 Начало таблицы
;      кодов директив
0: CMPB R0, (R4)+ Код совпадает?
      BEQ 1 Да — передать
;      управление
;      TSTB @R4 Конец таблицы?
;      BNE 0 Нет — продолжить
;      BR BEG Иначе кода в таблице нет, переход
;      к началу
1: SUB #DIR, R4 Номер кода
;      директивы
```

```

ASL   R4      Умножить на 2
JMP   @ADR-2(R4) Передать управ-
;      ление по таблице
;      меток в соответ-
;      ствии с номером
;      кода директивы,
;      с поправкой
DIR:  .B: 10, 22, 31, 32, 33.E
ADR:  .#M1, M2, M3, M4, M5

```

В данной программе использованы две таблицы. В первой, по метке DIR, — список кодов директив (для примера используются коды клавиш управления курсором). Введя директиву, мы ищем ее код в таблице. Если он найден, вычисляем его номер от начала таблицы. Но коды директив — байты, а адреса меток передачи управления — слова, поэтому удваиваем номер и вычисляем адрес соответственной метки в таблице ADR (как адрес начала таблицы плюс номер метки). По этой метке и передается управление. @ ADR — 2 появилось в программе потому, что после нахождения директивы R4 инкрементируется и указывает на адрес следующей директивы, и при обращении к метке мы вводим поправку. .E в конце таблицы DIR приводит адрес метки ADR к четному, а нулевой байт отмечает конец таблицы. Чтобы все было совершенно ясно, покажем, как можно организовать модули программы, к которым мы обращаемся таким образом. Пример приведен простейший — по нажатию клавиш-директив наша программа просто будет выдавать номера меток. Но точно так же можно построить любые модули. Итак, после программы допишем:

```

M1:   MOV     '1',R0   M1 - "стрелка влево"
      BR     TYP
M2:   MOV     '2',R0   M2 - "стрелка в
      BR     TYP       начало экрана"
M3:   MOV     '3',R0   M3 - "стрелка
      BR     TYP       вправо"
M4:   MOV     '4',R0   M4 - "стрелка вверх"
      BR     TYP
M5:   MOV     '5',R0   M5 - "стрелка вниз"
      BR     TYP
TYP:  EMT     16       Печать цифры на
;      экране
      BR     BEG       К началу
END

```

Приведенный способ управления интересен тем, что таблицы могут расширяться неограниченно по мере написания новых модулей программы.

.@MET — смещение к метке. При трансляции в слово по текущему адресу заносится разность между адресом метки и текущим адресом — *смещение к метке*. Как обычно, можно ввести в это смещение поправку, записав псевдооператор в виде .@MET+X или .@MET—X, где X — восьмеричное

число. Эта псевдокоманда широко используется для получения абсолютных адресов меток в перемещаемых программах. В самом деле, когда мы заносим абсолютный адрес метки, например, оператором MOV #MET,R4, то второе слово данной команды заносится в память один раз при компоновке, и его значение зависит от заданного адреса, по которому программа компоуется и в дальнейшем будет работать. Стоит только изменить адрес загрузки такой программы, и она работать не сможет, так как действительные адреса меток перестанут совпадать с полученными при компоновке. Иначе дело обстоит, если задано смещение: как бы мы ни перемещали программу, «расстояние» от текущего адреса до метки будет неизменным. Как пользоваться данной псевдокомандой? Предположим, нам нужно получить в каком-либо регистре абсолютный адрес метки MET. Запишем программу:

```

MOV   PC,R4      Занести текущий
;      адрес
ADD   (PC)+,R4   Привавить смещение
;      .@MET+2    Смещение с
;      поправкой на +2

```

Мы получили абсолютный адрес метки MET в регистре R4. Каким образом это делается? Вначале мы записали в R4 текущий адрес. Во второй строке прибавили к этому адресу смещение, полученное с помощью псевдооператора .@ в третьей строке. Мы добились этого, косвенно обращаясь через регистр PC, в котором в этот момент был адрес следующей команды, т. е. как раз нужной нам строки. Но в момент обращения к PC в первой строке мы записали адрес, на два меньший, чем адрес строки псевдооператора, и вынуждены исправить эту ошибку, внося поправку в смещение («+2»). То, что мы во второй строке не просто косвенно обращаемся к PC, а с инкрементом, служит важной цели — увеличить содержимое PC на 2 и таким образом «перескочить» строку с псевдооператором — передавать на нее управление нельзя, там не команда, а произвольное число. Пользуясь таким приемом, можно получить адрес метки независимо от того, находится она до или после псевдооператора. Как всегда в таких случаях, метка должна быть обычной, а не локальной. Возможны и иные приемы получения абсолютных адресов меток с помощью данного псевдооператора, но предложенный — самый экономичный по расходу памяти и быстродействию.

Все вышеперечисленные псевдооператоры могут быть записаны как в одну строку друг за другом в любой последовательности, так и в отдельных строках, но при этом надо учитывать, что псевдокоманды $\#$ и \textcircled{a} могут правильно работать только при трансляции их по четным адресам. В записанной последовательности операторов недопустимы пробелы и прочие символы, разрывающие ее. Такой символ либо ведет к ошибке, либо считается концом строки, и все, следующее за ним, транслятор игнорирует.

\textcircled{A} : — запись строки символов в коде КОИ-8. Код КОИ-8 — это «родной» символьный код БК-0010, включающий все его символы и коды управления в диапазоне 0—377. Иногда его называют кодом ASCII по аналогии с американским стандартом. Но, строго говоря, код ASCII — это совсем иной стандарт и никакого отношения к БК не имеет. Достаточно отметить, что он не содержит русских символов.

Так вот, данный псевдооператор позволяет записать в строку любые символы БК-0010; при трансляции их коды будут занесены в память начиная с текущего адреса, причем код каждого символа занимает один байт. Поскольку все, что следует за \textcircled{A} ., транслятор интерпретирует как символьный код, то этот псевдооператор может быть в строке только последним — после него недопустимы никакие операторы или комментарии. В версиях МИКРО.К пробелы заносятся в строку символов в любом месте обычным образом, в версиях МИКРО.С занесенные пробелы не всегда транслируются правильно, в частности игнорируются пробелы в конце строки. Если в строку символом нужно ввести управляющие коды (перевод строки, сброс экрана и т. п.), их можно задать с помощью псевдооператора \textcircled{B} .. Для самых распространенных кодов управления в МИКРО.10К последних версий введена возможность вставлять их в строку в виде символов $\textcircled{\wedge}$ (возведение в степень) и $\textcircled{\backslash}$ (обратная косая черта), они обозначают соответственно код 12 (перевод строки) и код 0. Примеры использования псевдооператора \textcircled{A} : можно найти в разделе, посвященном командным прерываниям ЕМТ. Он очень широко применяется для ввода в состав программ меню, текстовых сообщений, инструкций, таблиц кодов и т. п.

\textcircled{R} : — запись строки символов в коде RADIX-50. Как уже было сказано, при кодировании текста в коде КОИ-8 каждый символ занимает один байт. Это понятно, ведь используются все коды в диапазоне 0—377. Но иногда достаточно и меньшего количества кодов, например только цифр и латин-

ских символов. Простой расчет показывает, что если ограничиться только 40d символами, то в каждом машинном слове теоретически можно разместить до трех символов (если не верите, извлеките корень третьей степени из числа 65536d). Но такое размещение явно не пройдет «напрямую». В самом деле, если разделить 16 битов на три части, получится пять битов и один в остатке. В пяти битах можно закодировать не более 32d символов. Чтобы можно было полностью использовать информационную емкость машинного слова, прибегают к следующему приему. Весь кодируемый текст делят на триады символов, а затем в каждое слово записывают одну триаду по правилу:

$$\langle \text{значение} \rangle = ((C1 \times 50) + C2) \times 50 + C3,$$

где $C1$, $C2$, $C3$ — первый, второй и третий коды символов (все числа и вычисления — в восьмеричной системе). Число 40d равно восьмеричному 50, отсюда и название кода. Код позволяет при ограниченном наборе символов достичь полуторакратного повышения плотности упаковки. Применяется довольно редко, когда необходимо упаковать длинный текст при максимальной экономии памяти. Упаковка и распаковка кода RADIX-50 — не слишком простая задача, требующая большого времени по сравнению с обработкой текстов в формате КОИ-8, что тоже является препятствием для широкого применения RADIX-формата. Приведем таблицу восьмеричных значений кода RADIX-50.

Символ	Код RADIX-50
Пробел	0
A—Z	1—32
$\textcircled{\times}$	33
.	34
Резерв	35
0—9	36—47

Код 35 не используется, код 34 не задействован в МИКРО по техническим причинам. Псевдооператор \textcircled{R} ., как и \textcircled{A} ., должен быть последним в строке. Если в строке преобразуемых в код символов встретится символ, не входящий в состав кода, он считается концом строки. Если число преобразуемых символов не кратно трем, то последние символы автоматически дополняются до полной триады пробелами.

Нам осталось разобрать последний оператор — END. Он вполне может быть отнесен к псевдооператорам, так как не транслируется, а только сообщает транслятору о конце текста программы. Весь текст, следующий за ним, транслятор игнорирует. Этот оператор не обязателен в конце программы МИКРО.10К, как не обязателен и для неко-

торых иных версий ассемблера. Все же лучше его писать в конце текста программы во всех случаях. Помимо основного назначения может применяться, если нужно от-

транслировать только часть программы; в этом случае END временно ставится в конце транслируемой части.

Окончание следует.

А. ДИКОВ, А. КАЛАШНИКОВ, А. КУЛАКОВ

TermOS

Устройство расширения ОЗУ (УРОЗУ) (автор — А. М. Диков) предназначено для использования незадействованного адресного пространства БК-0010 с адресами 140000—157777 в качестве ОЗУ.

Программа, хранящаяся в УРОЗУ, запускается директивой Р (П) монитора и может быть испорчена только выключением питания или записью новой информации вместо старой. Перезапуск процессора переключателем ПУСК/СТОП информацию не искажает.

В УРОЗУ используются микросхемы К537РУ3А (16 шт.) или К537РУ8 (4 шт.) и несколько недефицитных микросхем серий 561 или 564. Микросхемы потребляют очень малый ток, поэтому устройство питается от БК-0010. Подключается оно к разъему «общая шина».

УРОЗУ доступно для повторения радиолюбителями, имеющими опыт работы с цифровыми микросхемами, не требует наладки. В системе TermOS оно предназначено для хранения основной программы и каталогов файлов магнитных лент. Очень удобно использовать его также для системных программ (редакторов, ассемблеров, отладчиков, баз данных). Это значительно увеличивает размер обрабатываемых файлов (текстов, листингов, данных и т. д.). Такие программы нами уже используются.

Магнитофон для работы в системе TermOS должен удовлетворять следующим условиям:

управление режимами воспроизведение, запись, перемотка влево и вправо, останов должно осуществляться электрическими импульсами — логическими 0 и 1;

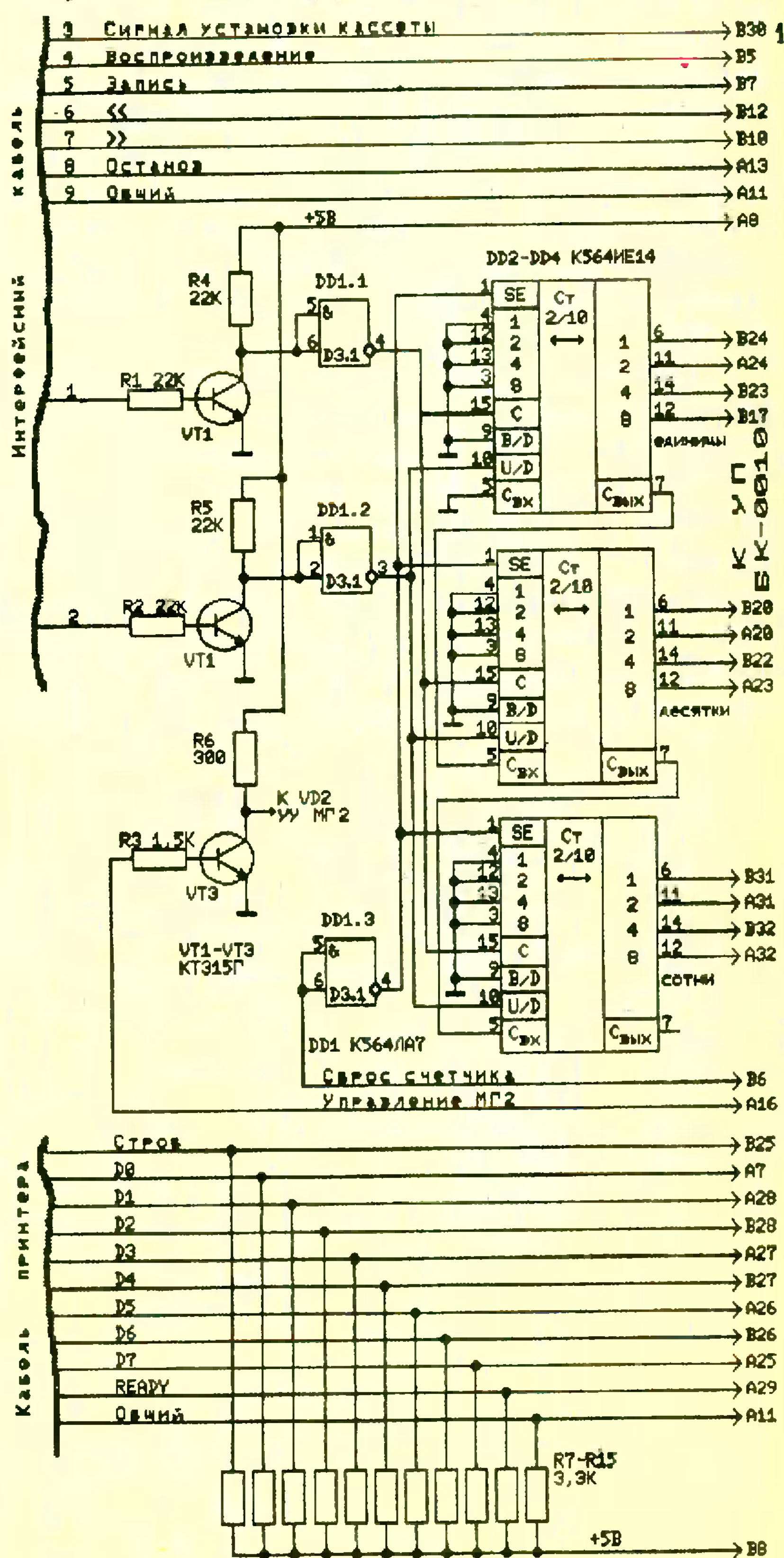
должен иметься датчик импульсов для счетчика ленты;

в схеме магнитофона должен иметься сигнал, указывающий направление движения ленты (0 — перемотка назад, 1 — остальные режимы);

Окончание. Начало см.: Информатика и образование. 1991. № 2.

должен иметься датчик нахождения кассеты в магнитофоне.

Управление магнитофоном осуществляется при помощи интерфейсного кабеля со встроенными трехдекадным счетчиком ленты



Выходной регистр			Входной регистр		
Разряд	Контакт	Сигнал	Разряд	Контакт	Сигнал
0	A16	Управление вторым магнитофоном	0	B24	Единицы счетчика
1	A13	Останов МГ1	1	A24	»
2	B12	Перемотка влево	2	B23	»
3	B10	Перемотка вправо	3	B17	»
4	B5	Воспроизведение	4	B20	Десятки счетчика
5	B7	Запись	5	A20	»
6	B6	Сброс счетчика	6	B22	»
7	A7	D0 принтера	7	A23	»
8	A28	D1 принтера	8	B31	Сотни счетчика
9	B28	D2 принтера	9	A31	»
10	A27	D3 принтера	10	B32	»
11	B27	D4 принтера	11	A32	»
12	A26	D5 принтера	12	B30	Кассета в МГ?
13	B26	D6 принтера	13	A. 9	READY принтера
14	A25	D7 принтера			
15	B25	Строб принтера			

(размещение счетчика в магнитофоне потребовало бы увеличения числа соединительных проводов между ним и БК с 9 до 20).

Кабель интерфейса со встроенным счетчиком ленты подключается к параллельному порту БК-0010. Назначение сигналов, поступающих на порт, приведено в таблице. Электрическая схема счетчика показана на рис. 1.

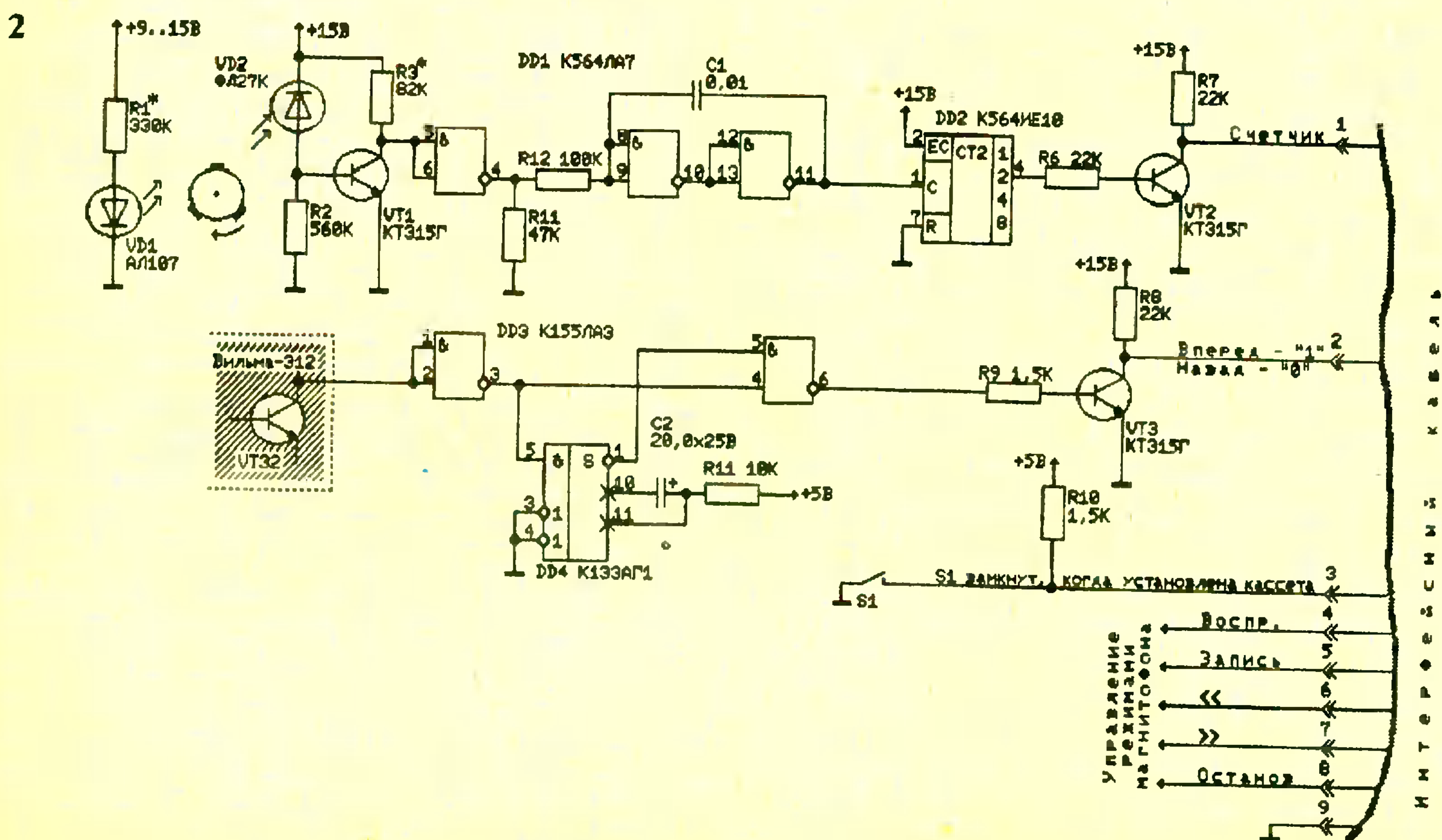
Счетчик ленты собран на трех двоично-десятичных реверсивных счетчиках К564ИЕ14 (их можно заменить на К561ИЕ14), включенных в режим десятичного счета подачей на вход В/D уровня логического 0. Счетные импульсы с датчика в МГ1 через буферный каскад на VT1 и инвертор DD1.1 подаются на счетные выходы С счетчиков, соединенных последовательно. Сигнал направления счета (прямого или обратного)

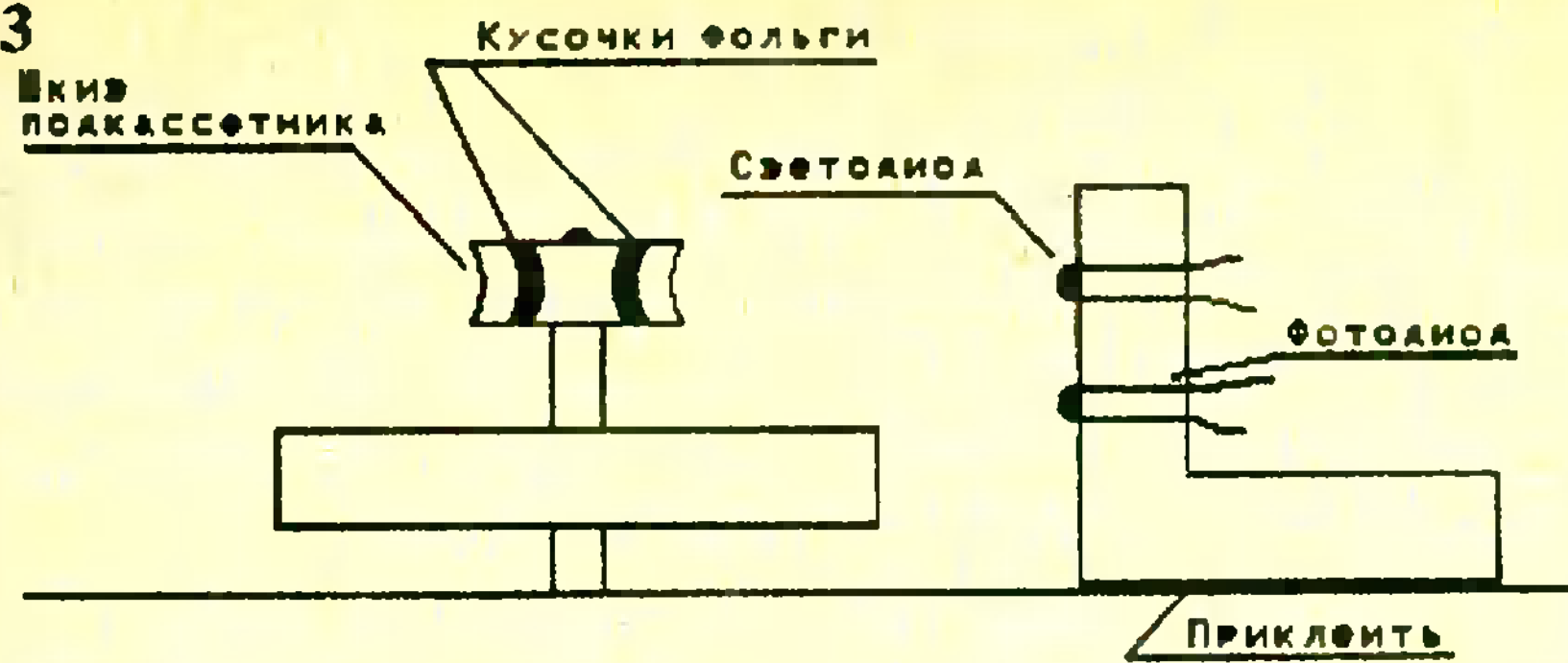
с МГ1 через буферный каскад на VT2 и инвертор DD1.2 подается на входы U/D. Сброс счетчика происходит при поступлении сигнала с порта БК через инвертор DD1.3 на входы SE. Коррекция счетчика осуществляется программно.

Каскад на VT3 предназначен для буферизации сигнала управления включением МГ2.

Питание на схему подается с разъема порта БК, для чего в некоторых экземплярах необходимо запаять перемычку, подающую +5 В на разъем порта. Микросхемы серии 564 можно заменить на микросхемы серии 561.

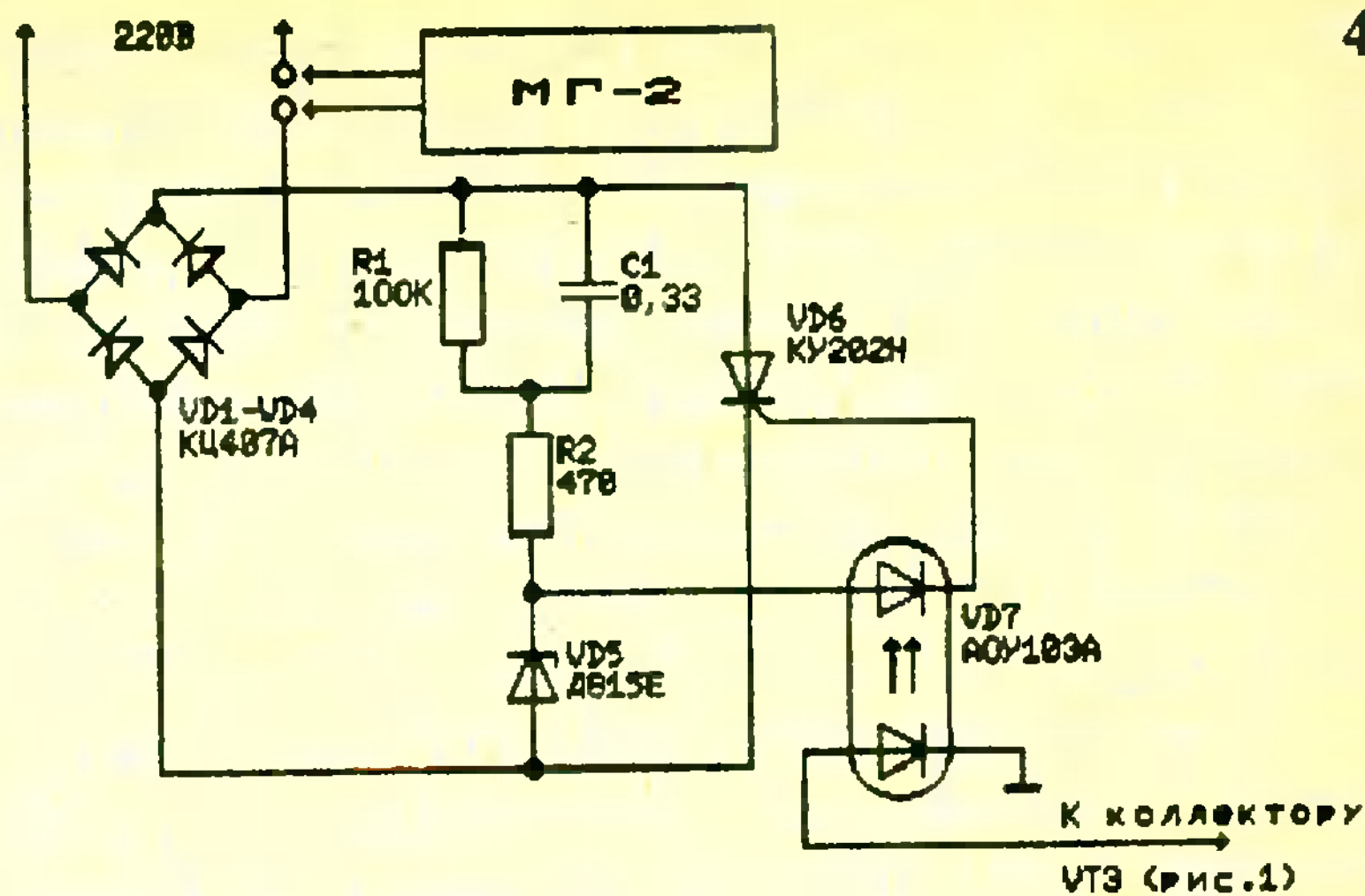
В используемый нами магнитофон «Вильма-312 стерео» были внесены следующие изменения и дополнения (рис. 2):





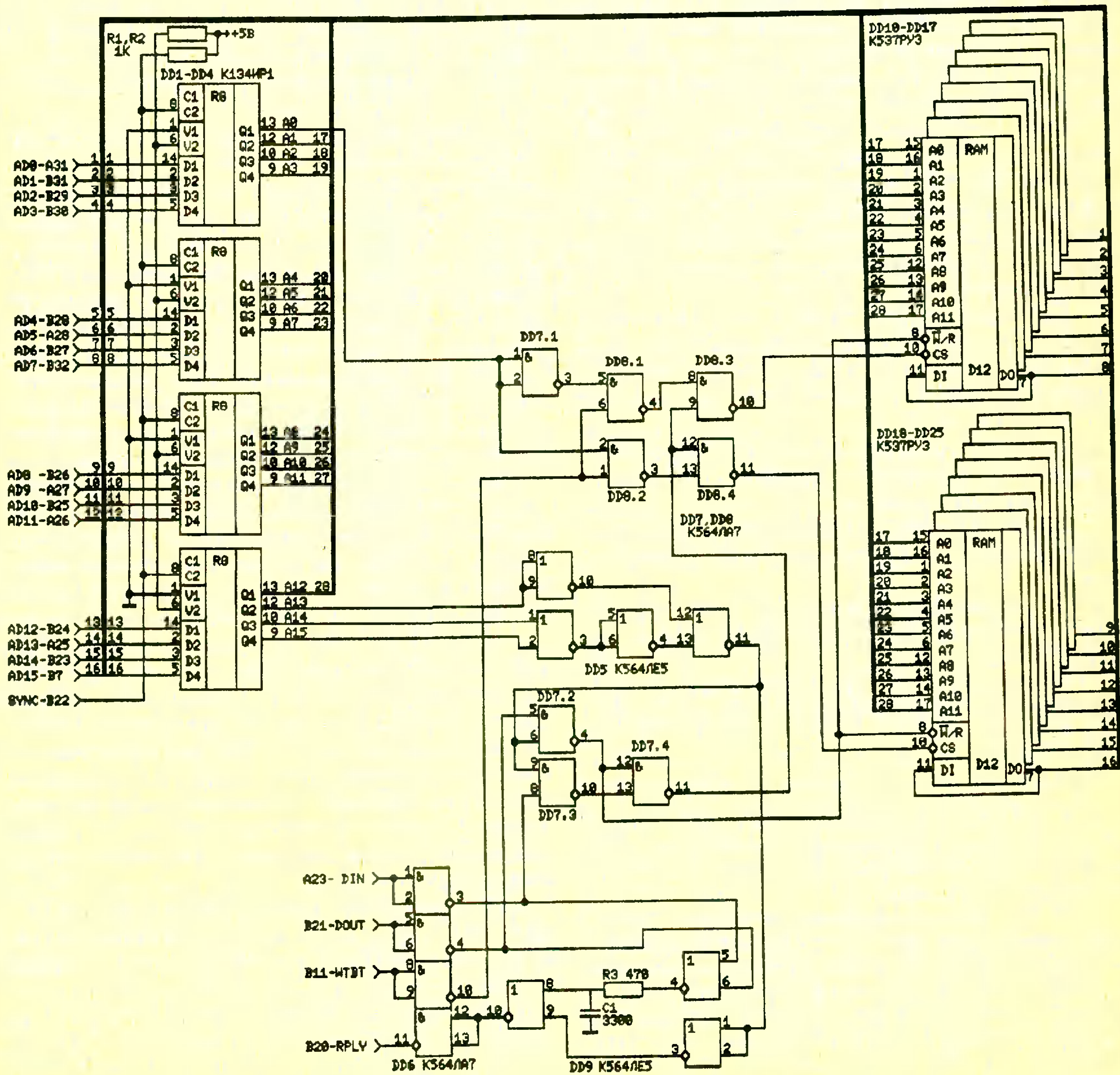
для исключения включения режима пауза при включении режима запись перерезан печатный проводник на плате блока питания и управления (A2), подходящий к ножке 10 триггера K155TM2 (DD2.1);

для исключения ошибки счетчика из-за «выбега» ленты при выключении режима перематка влево (счетчик уже переключен на прямой счет, а лента еще движется назад) на одновибраторе 155АГ1 собрана схема за-



держки выключения сигнала *вперед-назад*; изготовлен датчик импульсов.

Устройство датчика импульсов показано на рис. 3. Шкив, жестко связанный с приемным подкассетником, зачерняется черной краской. Затем на него наклеиваются три полосы алюминиевой фольги шириной по



3 мм. В приклеенном рядом кронштейне укрепляются светодиод АЛ107 и фотодиод ФД27К. Каскады на VT1 и DD1 формируют прямоугольные импульсы, которые подаются на счетчик DD2, обеспечивающий необходимый коэффициент деления (в приведенной конструкции при установке кассет МК-60 и МК-90 концу ленты соответствуют показания счетчика около 600 и 900 соответственно).

Сигнал направления движения ленты берется с коллектора VT32. В других магнитофонах этот сигнал можно снять, например,

со схемы индикации режимов работы.

Устройство управления вторым магнитофоном позволяет управлять МГ2, не имеющим электронного управления, не внося никаких изменений в его схему. Управление производится включением и отключением напряжения питания магнитофона.

Кроме того, это устройство можно использовать для управления какой-либо мощной нагрузкой в программах пользователя.

Электрическая схема устройства показана на рис. 4.

Оптимальный драйвер магнитофона для БК

Драйвер магнитофона, «прошитый» в ПЗУ БК-0010 (в дальнейшем именуемый штатным), обеспечивает скорость обмена информацией 1200 бит/с. Для записи информации на МЛ используется метод широтно-импульсной модуляции* с побитовой синхронизацией**. Степень надежности считывания информации вряд ли можно считать высокой даже при использовании двух экземпляров записи файла, особенно для достаточно длинных программ. Нельзя ли улучшить программ-драйвер?

Проанализируем требования, которым должен удовлетворять оптимальный драйвер магнитофона для БК-0010 (с учетом использования дешевых и несовершенных моделей магнитофонов).

1. Устойчивость работы при больших нелинейных искажениях сигнала, поступающего с магнитофона.

2. Желательно сохранение работоспособности при различных скоростях движения МЛ.

3. Индифферентность к возможному инвертированию импульсов, получаемых от магнитофона.

4. Сохранение работоспособности при незначительных колебаниях скорости движения МЛ.

* Широтно-импульсная модуляция — способ представления информации, при котором информационное значение импульса определяется его длительностью. — *Примеч. ред.*

** Побитовая синхронизация — способ передачи информации, при котором каждому информационному импульсу предшествует так называемый синхроимпульс, не несущий информационной нагрузки, а служащий для уведомления о том, что через стандартный интервал времени последует информационный импульс. — *Примеч. ред.*

5. Максимальная плотность записи на МЛ (при фиксированной максимальной частоте записываемого сигнала).

6. Максимальная надежность считывания.

7. Устойчивость к помехам (наводкам) периодического характера.

1. Из всех видов магнитной записи цифровой информации в условиях больших нелинейных искажений наилучшим является метод широтно-импульсной модуляции. Он, в частности, положен в основу штатного драйвера, причем каждому импульсу предшествует синхроимпульс, т. е. используется побитовая синхронизация. Но нужна ли она? Быстродействие процессора К1801ВМ1 вполне достаточно, чтобы, занеся после декодирования очередного информационного импульса бит в ОЗУ и вернувшись на чтение сигнала с магнитофона, «потерять» не слишком большую часть начала следующего импульса. Можно было бы сделать поправку на время занесения бита, но она очень мала, и учитывать ее не обязательно.

2, 3. Этим требованиям штатный драйвер удовлетворяет вполне, так как является настраиваемым. По расположенной в начале файла настроечной последовательности, заканчивающейся маркером, вырабатывается константа, с помощью которой осуществляется декодирование, а маркер служит указателем начала оглавления файла, а также для определения характера чтения (прямое или инверсное).

Настройка драйвера по скорости движения МЛ известна, хотя и используется в основном обладателями катушечных магнитофонов; менее известен факт возможного инвертирования сигнала при использовании фильтров или разных выходов магнитофона.

4. Защищенность от высокого коэффи-

циента детонаций, характерного для дешевых магнитофонов, обеспечивается большой разницей длительностей информационных импульсов. В штатном драйвере единица кодируется вдвое более длинным, чем ноль, импульсом. Практика показывает, что такое соотношение длительностей оптимально.

5. Этот пункт наиболее важен. Задача (с учетом вышесказанного) формулируется так: выбрать наиболее информативный способ кодирования при условии, что алфавит кодируется кратными по длительности импульсами, а длительность самого короткого импульса фиксирована.

Пусть M — число битов, кодируемых одним импульсом (например, в штатном драйвере $M=1$: длинный импульс кодирует 1, короткий — 0; если используются импульсы четырех длительностей: $T, 2T, 3T, 4T$, то каждый кодирует два бита, $M+2$; например, импульс длиной T может обозначать 00, $2T$ — 01, $3T$ — 10, $4T$ — 11). Тогда для кодирования всего алфавита потребуется $K=2^M$ импульсов с периодами $T, 2T, \dots, KT$. Средняя длительность импульса T_C будет равна $T_C = (T + 2T + \dots + KT) / K = (K + 1)T / 2$, а среднее время T_1 , затрачиваемое на передачу одного бита — $T_1 = T_C / M = (K + 1)T / 2M = (2^M + 1)T / 2M$. Эта функция принимает максимальное значение при $M=2$, при этом $T_1 = 1,25T$.

Итак, для обеспечения максимальной скорости передачи информации при данных условиях необходимо кодировать одним импульсом два бита. А сколько времени тратит на передачу бита штатный драйвер? При каждом информационном импульсе (длительностью T для нуля и $2T$ для единицы) им передается еще и синхроимпульс длительностью T , так что $T_1 = (T + T + T + 2T) / 2 = 2,5T$: вдвое больше оптимума! А ведь оптимальный драйвер не предъявляет дополнительных требований ни к магнитофону, ни к МЛ: если штатный драйвер использует импульсы длительностью T и $2T$ и маркерный импульс длительностью $4T$, то оптимальный — импульсы длительностью $T, 2T, 3T, 4T$.

6. Надежность оптимального драйвера выше, чем у штатного, уже хотя бы потому, что запись того же массива в оптимальном варианте вдвое короче и содержит вчетверо меньше импульсов (что ясно из предыдущего). Повысить надежность еще больше можно, придав файлу блочную структуру и сопроводив каждый блок номером, таблицей кодирования, контрольной суммой (см. ниже).

7. Большинство пользователей встречались с ситуацией, когда при считывании с МЛ драйвер, вместо того чтобы искать файл с за-

данным именем, начинает распечатывать немыслимые имена несуществующих файлов. Причина в том, что в магнитофонном кабеле имеется периодическая помеха (например, наводка от сети), по которой и происходит настройка драйвера. Для нейтрализации этого явления можно, например, ввести специальный ключевой символ в один из байтов оглавления файла; его отсутствие будет сигнализировать о необходимости прекратить чтение и повторить настройку.

В соответствии с результатами этого анализа был написан драйвер магнитофона для БК-0010 HELP7. Он использует для записи информационные сигналы длительностью $T, 2T, 3T, 4T$ и маркерный импульс длительностью $4T$.

При записи файла драйвер записывает сначала настроечную последовательность, затем блок параметров, в который входят имя файла, адрес его загрузки, длина и контрольная сумма файла, метка &, идентифицирующая файлы в формате HELP7. Сам файл разбивается на блоки длиной 256 байтов каждый (кроме, может быть, последнего). Каждый блок предваряется четырехбайтовым оглавлением.

Первый байт оглавления — ключ декодирования. В нем диады 00, 01, 10, 11 располагаются в порядке возрастания частоты их появления в блоке. Первая из диад ключа кодируется в данном блоке импульсом длиной T , вторая — $2T$ и т. д. Благодаря этому обеспечивается минимальное время передачи блока. В частности, блок из одних нулей передается в четыре, а из одних единиц — в шесть раз быстрее, чем штатным драйвером.

Второй байт — номер блока. Его наличие позволяет обеспечить независимое чтение блоков; зная номер, драйвер может определить положение блока в ОЗУ, следовательно, блоки могут считываться в любом порядке.

Третий и четвертый байты — контрольная сумма блока, сложенная с контрольной суммой всего файла. Такое сложение позволяет «сбрасывать» блоки, не принадлежащие данному файлу, как прочтенные с ошибкой (вычисленная при считывании контрольная сумма блока, сложенная с контрольной суммой файла из блока параметров, не совпадает с контрольной суммой из оглавления блока). Кроме того, увеличивается надежность чтения благодаря тому, что контрольную сумму имеет каждый блок.

При чтении HELP7 сперва ищет на МЛ очередное оглавление блока и определяет по номеру блока, был ли он ранее считан без ошибки. Если да — осуществляется поиск следующего блока; если нет — блок

считывается и помещается на соответствующее место в ОЗУ. Таким образом, если какие-то блоки не считываются, а файл записан на МЛ дважды, их можно считать со второго экземпляра файла. В результате значительно повышается надежность хранения информации.

На практике HELP7 при высокой надежности чтения-записи обеспечивает скорость обмена до 6000 бит/с; обычно используется на скорости 4000 бит/с.

Ниже приводятся распечатки программ записи и чтения массива, достаточно полно иллюстрирующих изложенные идеи. Хотя в них не отражен ряд моментов работы HELP7 (определение постоянных чтения, блочная передача информации, работа с ЦАП и др.), они могут послужить базой для полезных и интересных экспериментов даже неопытному программисту.

Программа записи массива

Содержимое регистров:

R1 - адрес записываемого на МЛ массива
R2 - его длина в байтах
R3=177716

```

M0: MOV B (R1)+,R0 ; Записать в R0 оче-
; редной байт
M1: MOV #4,R4 ; Один байт - это
; четыре диады
ASR R0 ; Арифметический
BCS A1 ; сдвиг вправо
; Анализ первого би-
; та диады
ASR R0 ; Анализ второго би-
BCS A2 ; та диады
; Передается диада
; "00"
MOV #160,@R3 ; Установка высокого
; уровня
MOV T00,R5 ; (T00) - длитель-
; ность импульса
; "00"
0M: SOB R5,0M ; Задержка
MOV #20,@R3 ; установка низкого
; уровня
1M: SOB R5,1M
BR KN ; Передается диада
; "10"
A2: MOV #160,@R3
MOV T10,R5
2M: SOB R5,2M
MOV #20,@R3
MOV T10,R5
3M: SOB R5,3M
BR KN
A1: ASR R0
BCS A3 ; Анализ второго би-
; та диады
; Передается диада
; "01"
MOV #160,@R3
MOV T01,R5
4M: SOB R5,4M
MOV #20,@R3

```

```

MOV T01,R5
5M: SOB R5,5M
BR KN ; Передается диада
; "11"
A3: MOV #160,@R3
MOV T11,R5
6M: SOB R5,6M
MOV #20,@R3
MOV T11,R5
7M: SOB R5,7M
KN: SOB R4,M1 ; Цикл по диадам
SOB R2,M0 ; Цикл по длине
RTS PC

```

Программа чтения массива

Содержимое регистров:

R1 - адрес, с которого размещается считываемый с МЛ массив
R2 - длина массива
R3=177716
R5=40

Импульсы длительностью T, 2T, 3T, 4T кодируют диады 00,01,10,11.

Постоянные чтения: 1,5T; 2,5T; 3,5T.

```

M0: MOV #4,R0 ; Чтение четырех диад
M1: CLR R4 ; Очистка R4
1M: INC R4 ; R4=R4+1
BIT R5,@R3 ; Тестирование 5-го
; разряда регис-
; тра 177716
BEQ 1M
2M: INC R4
BIT R5,@R3
BNE 2M
CMPB R4,C2 ; (C2)=2,5T. Сравнение
; со второй постоянной
; чтения
BNI M2 ; Длительность импуль-
; са больше 2T
CMPB R4,C1 ; (C1)=1,5T. Сравнение
; с первой постоянной
; чтения
BNI M3 ; Длительность импуль-
; са равна 2T
; Длительность импуль-
; са равна T
; Запись диады "00" в
; память
CLC
RORB @R1
CLC
RORB @R1
BR KN ; Запись диады "01" в
; память
M3: SEC
POPB @R1
CLC
RORB @R1
BR KN
M2: CMPB R4,C3 ; (C3)=3,5T. Сравнение
; с третьей постоянной
; чтения
BNI M4 ; Запись диады "10" в
; память
CLC
RORB @R1
SEC

```

```

RORB @R1
BR KN
; Запись диады "11" в
; память
M4: SEC
RORB @R1
SEC
RORB @R1
KN: SOB R0, M1 ; Цикл по диадам

```

```

INC R1 ; Следующий байт
SOB R2, M0 ; Цикл по длине
RTS PC

```

Заинтересовавшиеся этой темой могут написать автору по адресу: 194017, Ленинград, ул. Рашетова, 11, кор. 2 кв. 16.

А. МОЛЧАНОВ

Подключение принтеров «Электроника МС-6312» и «Электроника МС-6313» к БК

В московском магазине-салоне «Электроника» появились в продаже принтеры моделей МС-6312, МС-6313, предназначенные для печати алфавитно-цифровой и графической информации. Обе модели представляют боль-

шой интерес уже потому, что первыми появились в свободной продаже.

В табл. 1 приведены их основные технические характеристики, анализ которых позволяет сделать следующие выводы.

Таблица 1 **55**

	МС-6312	МС-6313
Тип УВП	термоструйное	ударное, знаковосинтезирующее
Тип бумаги	ГОСТ 6656-76	ГОСТ 6656-76
Формат бумаги	А4 или рулонная	А4 или рулонная
Тип красконосителя	термоструйная печатающая головка	лента красящая синтетическая
Ресурс	800—1200 страниц	1000 страниц
Режим печати:		
символьный	есть	есть
графический	»	»
Средства для диагностики:		
тест шрифтов	есть	есть
тест графики	нет	»
тест косой печати	»	»
Число символов в строке	40—120	40—160
Шаг строки, мм	2,12—4,23	2,12—12,7
Скорость печати, зн/мин	150	100
Матрица знакоместа, точек	16×12	9×12, 18×24
Качество шрифта:		
качественный	есть	есть
черновой	нет	»
Типы шрифтов:		
нормальный	есть	есть
двойной	»	»
высокий	нет	»
подчеркивание	есть	»
верхние индексы	»	»
нижние индексы	»	»
Кодирование	КОИ-8	КОИ-7, КОИ-8
Количество наборов символов	3	7
Совместимость со стандартом Epson	есть	нет
Тип интерфейса	ИРПР, ИРПР-М	ИРПР, ИРПР-М, С2
Напряжение питания, В	220±10%	220±10%
Потребляемая мощность	30 Вт	90 Вт
Габаритные размеры, мм	55×166×277	100×400×460
Масса, кг	2,3	8
Цена, руб.	1525	1675
Завод-изготовитель	ПО «Радий», Москва	завод «Кентавр», Смоленск

Устройство вывода информации печатающее (УВИП) МС-6312:

достоинства — малые вес и габариты, бесшумная высокоскоростная печать, по системе кодов практически полная совместимость со стандартом Epson ESC/—P80

недостаток — в качестве красконосителя используется специальная печатающая головка, которая не подлежит заправке, а заменяется новой; стоимость одной головки — 25 рублей.

Отметим также, что инструкция по эксплуатации УВИП МС-6312 недостаточно полна и не лишена неточностей: например, в команде ESC C неверно указан десятичный код, неверно описана функция команды. Отсутствуют примеры применения команд.

Устройство вывода печатающее (УВП) МС-6313:

достоинства — широкий набор графических символов, доступность красящей ленты, высокое качество печати, наличие последовательного интерфейса, нетребовательность к типу бумаги;

недостаток — несовместимость по системе кодов со стандартом Epson ESC/—P80.

Принтеры могут работать совместно с компьютером БК-0010(01), однако для МС-6313 необходимо несколько изменить подпрограмму передачи кодов от компьютера к УВП в плане формирования положительного фронта импульса STROBE. Например, чтобы обеспечить функционирование команды nW программы-отладчика ОТЛ12, необходимо подпрограмму передачи кодов переписать в следующем виде:

```
037612 BIS #400,R0
037616 TST @#177714
037622 BEQ 37616
037624 MOV R0,@#177714
037630 SUB #400,R0
037634 MOV R0,@#177714
037640 RTS R7
```

Для более полного использования возможностей принтеров разработаны драйверы MS6312.SAV и MS6313.SAV. Построенные по сходным алгоритмам, они различаются наборами задающих режимов. В качестве примера приводится описание драйвера MS6313.SAV.

Драйвер предназначен для печати текстовых файлов, созданных редакторами EDASP, МИКРО.10К, МИКРО.11К, на печатающем устройстве МС-6313.

Драйвер можно загружать в любое место оперативной памяти, он занимает около 2К байт ОЗУ и работает в режиме расширенной памяти. Если загрузка прошла нормально, программа переходит к выяснению готовности принтера. В случае неготовности на экране появляется сообщение «ПЕЧАТЬ НЕ

ГОТОВА». При приведении принтера в состояние готовности к приему информации на экране появится приглашение к работе в виде головного меню:

- 1 — СБРОС
- 2 — ЧТЕНИЕ
- 3 — ПЕЧАТЬ
- 4 — ТЕКСТОВЫЙ РЕЖИМ
- 5 — ГРАФИЧЕСКИЙ РЕЖИМ

Для выбора режима работы необходимо нажать соответствующую цифровую клавишу.

Режим СБРОС предназначен для установки принтера в исходное состояние, действует аналогично включению питания. Сигнал может быть заблокирован переключателем SA3.5 принтера. Для исключения ошибочного сброса запрашивается подтверждение.

Режим ЧТЕНИЕ предназначен для загрузки файла с целью его последующей распечатки.

ПЕЧАТЬ обеспечивает вывод на принтер предварительно загруженного текстового файла с параметрами печати, заданными режимами 4, 5. При распечатке файла в левой верхней части листа распечатывается его имя.

Текстовый режим предназначен для вызова меню задания режимов управления текстовой печатью.

Графический режим предназначен для задания плотности графической печати в диапазоне 577—1921 ударных позиций в строке. Выход из графического режима в текстовый осуществляется по нажатию клавиши 0.

В текстовом режиме есть свое меню:

- 0 — ВЫХОД
- 1 — ФОРМАТ
- 2 — ТАБУЛЯЦИЯ
- 3 — ШАГ СТРОКИ
- 4 — ШАГ ЗНАКОВ
- 5 — КАЧЕСТВО ШРИФТА
- 6 — АТТРИБУТЫ ШРИФТА

Для выбора управления необходимо нажать соответствующую клавишу.

0 — выход в головное меню.

1 (формат) позволяет задать число строк на странице десятичным числом от 0 до 255. При задании нулевого значения принтер не следит за форматом, но контролирует наличие бумаги.

2 (табуляция) позволяет задать левую границу страницы, при этом активная позиция печати перемещается вправо на число знаков, равное $N \times 8$, где $N \langle \text{ВВОД} \rangle$ — задаваемое число позиций табуляции.

3 (шаг строки) позволяет задать шаг строки (от 2,12 до 12,7 мм). Для выбора необходимо нажать соответствующую клавишу.

Таблица 2

МС-6313		БК-0010.01	
Сигнал	Контакт	Контакт	Сигнал
DATA0	3	A16	ВД00
DATA1	5	A13	ВД01
DATA 2	7	B12	ВД02
DATA 3	9	B10	ВД03
DATA4	11	B5	ВД04
DATA5	13	B7	ВД05
DATA7	17	A7	ВД07
STROBE	23	A28	ВД08
BUSY	21	B31	ВВ08
ASCNLG	25	A23	ВВ07
INIT	27	B28	ВД09
OY	4, 6, 8, 10, 12 14, 16, 18, 20 22, 24, 26, 28	A11, A18 B11, B18 A19, B19	Общий

4 (шаг знаков) позволяет задать шаг знаков (число символов в строке) в диапазоне от 1,27 до 5,08 мм (40—160 симво-

лов). Переключатель SA3.3 должен быть в положении «включено».

5 (качество шрифта) позволяет задать черновиковый либо качественный шрифт, похожий на шрифт пишущей машинки.

6 (атрибуты шрифта) — задание дополнительных вариантов шрифта. При нажатии клавиши 0 осуществляется отмена всех атрибутов.

Для задания режимов работы драйвера рекомендуется следующий порядок:

сброс; чтение; задание режимов печати; печать; возврат либо на чтение, либо на задание режимов печати.

Принтер функционирует от параллельного интерфейса ИРПР по ОСТ 11.305.917-84. Схема его подключения к БК приведена в табл. 2. Переключатели принтера должны быть установлены в следующее положение: SA1 — 01001101; SA2 — 11000000; SA3 — 11011001.

А. БАРСУКОВ

57

Программатор микросхем ППЗУ 558PP3 для БК-0010

Законное раздражение многих пользователей БК вызывает неудобство работы с магнитофоном: малая скорость обмена, сбои, трудность поиска нужной информации на МЛ... Попытки решить проблему «в лоб» пока не дали достаточно хороших результатов; а ведь есть удобный обходной путь — использование в качестве внешней памяти постоянных перепрограммируемых запоминающих устройств (ППЗУ). Это дает возможность всегда иметь «под рукой» наиболее часто используемые программы, варьировать их состав, практически забыть о сбоях.

В предлагаемом программаторе ППЗУ используются микросхемы с электрическим стиранием информации К558PP3. Программатор работает в составе БК под управлением специальной программы, позволяющей записывать в ППЗУ и стирать необходимую информацию. Чтение из ППЗУ осуществляется непосредственно, как из ОЗУ.

Технические характеристики программатора:

- время выборки данных из ППЗУ 170 нс
- время стирания 20 с
- время записи одного слова 50 мс
- напряжение программирования 24 В
- напряжение стирания 18 В
- информационная емкость 16К байт
- начальный адрес ППЗУ 120000

напряжение питания (внешний источник) 27—30 В

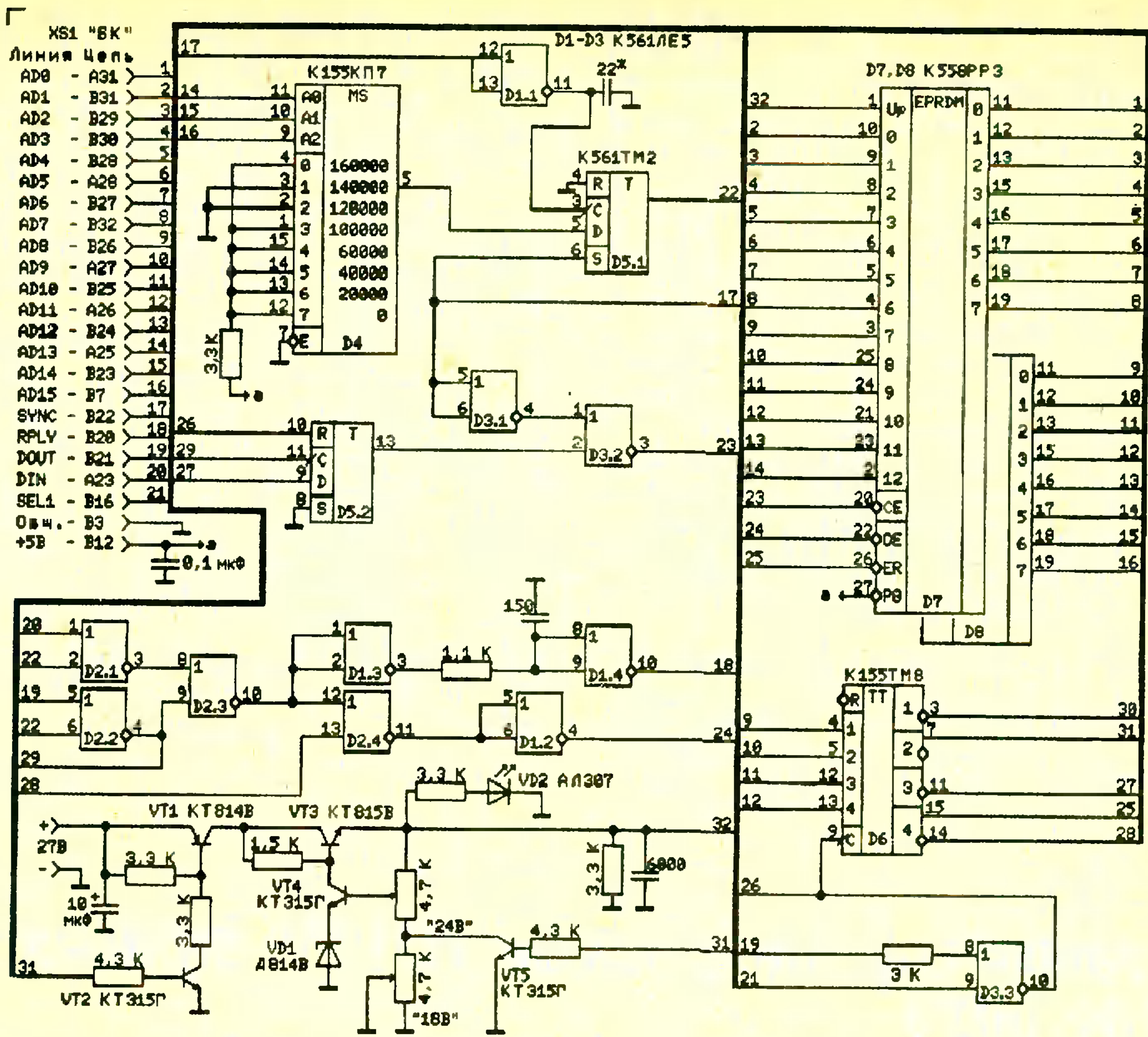
интерфейс — совместимый с общей шиной БК-0010 (.01)

Для начального адреса ППЗУ могут быть установлены значения 0, 20000, 40000, 60000, 100000, 140000.

Из входящих в программатор шести микросхем лишь три необходимы в случае использования его как модуля ПЗУ (т. е. только для чтения).

Программатор состоит из двух основных блоков: ППЗУ (две микросхемы К558PP3) и устройства управления ими. Последнее формирует сигналы выборки ПЗУ CS, выборки адреса SE, выборки данных SO, а также напряжения программирования и стирания.

Схема выборки собрана на мультиплексоре D4 и триггере D5.1. Перемычки на входе D4 служат для установки диапазона адресов, в которых работает ППЗУ. Если адрес на входах D4 лежит в заданном диапазоне, то D4 вырабатывает импульс, который стробируется сигналом SYNC и устанавливает триггер D5.1. Сброс триггера происходит по его входу S задним фронтом сигнала SYNC. На выходе D5.1 формируется сигнал CS, под управлением которого и происходит любой обмен с ППЗУ.



58

Вторая половина триггера, D5.2, управляет прохождением сигнала SYNC на вход CE ППЗУ через элементы D3.1 и D3.2. D5.2 разрешает сигнал CE, если в регистре управления D6 установлен бит разрешения записи. Сбрасывается триггер при очередном обращении к регистру управления. Такое решение позволяет программно затянуть импульс CE до длительности в 50 мс, требуемой при программировании ППЗУ, или вообще отключить его, что необходимо при стирании.

Элементы D2 и D1 служат для получения сигналов стробирования данных OE и ответа RPLY. Логика схемы такова, что данные стробируются на выводах данных ППЗУ только при наличии сигнала выборки CS. Сигнал OE может быть заблокирован одним из разрядов регистра управления — это нужно при стирании. RPLY формируется элементами D1.3 и D1.4, представляя собой сигнал DIN или DOUT, возвращаемый процессору с задержкой 150—200 нс.

Формирование необходимой временной диаграммы и коммутацию питающего напряжения выполняет регистр управления D6. Назначение его разрядов: 1 — включение напряжения 24 В; 2 — включение напряжения 18 В; 3 — установка режима записи; 4 — установка режима стирания. Адрес регистра — 177716. Это системный регистр БК,

в программаторе используется его старший байт, не задействованный в БК. Начальная установка регистра (режим чтения) происходит при включении БК или при перезапуске монитора.

В схеме использовано несколько транзисторов, задача которых — стабилизация и коммутирование напряжений программирования и стирания.

Главной частью схемы являются ППЗУ K558PP3. Следует отметить особенность их установки в шину БК: каждое ППЗУ выдает один байт (D7 — младший, D8 — старший), поэтому адресные входы ППЗУ подключены к шине начиная не с AD0, а с AD1 (обмен по словам).

Для управления работой программатора используется специальная управляющая программа «558PP3». Она позволяет:

- стирать ППЗУ;
- записывать информацию в ППЗУ;
- проверять ППЗУ на чистоту;
- проверять качество записи в ППЗУ;
- загружать информацию в ППЗУ с магнитофона;

программировать ППЗУ в ручном режиме.

Все эти функции представлены в меню программы. На те из них, которые оказывают необратимое воздействие, запрашивается дополнительное подтверждение. Объ-

ем программы — 1,5К байт, объем рабочего буфера — 26К, язык программирования — ассемблер МИКРО10.К.

Конструктивно программатор удобно выполнить в виде приставки, собираемой в корпусе от блока МСТД, и подключать к БК через разъем системной шины ХТЗ.1. На верхней крышке приставки можно расположить разъемы для подключения ППЗУ и светодиода индикации программирующего напряжения.

За время подготовки данной статьи к печати программатор был успешно применен автором при разработке дисковой операционной системы для БК-0010 РАМОНГ.

За справками можно обращаться по адресу: 235410, Шяуляй, Севастополе, 9, кв. 44.

Автор благодарен В. И. Джиган и С. Г. Лужецкому (см.: Микропроцессорные средства и системы. 1988. № 1. С. 85—88) за идею схемы.

А. ЕРМАКОВ

Увеличение символов на экране

Многим программистам-любителям знакома ситуация, когда возникает желание подписать рисунок или схему крупными буквами. Увы, два шрифта БК не позволяют этого сделать, и приходится рисовать буквы оператором DRAW, засоряя ОЗУ.

Предлагаемые программы SCR12 и SCR24 позволяют увеличить символы в 1×2 и 2×4 раза соответственно. Работа с ними происходит так.

1. Загрузить в БК код программы SCR12.BIN (или SCR24.BIN). Это можно сделать из Бейсика операторами DATA и READ или с МЛ оператором BLOAD; программы написаны в позиционно-независимом коде, т. е. адрес загрузки можно выбрать произвольно.

2. Определить машинную подпрограмму оператором

```
DEF USR=<адрес>
```

3. Обязательно выровнять экран:

```
PRINT CHR$(140);CHR$(140)
```

4. Обратиться к подпрограмме оператором

```
R$(R)=USR(R)
```

передав ей значение символьной величины $R \text{ } \text{ } = \text{«XYТекст»}$, где X — номер позиции в строке, начиная с которой будет выводиться Текст, Y — номер строки, в которой будет выводиться Текст, Текст — произвольный набор символов Бейсика.

Например, оператор

```
R$(R)=USR("65ПРИВЕТ")
```

выведет слово ПРИВЕТ, начиная с шестой позиции пятой строки экрана.

В нашем кабинете информатики разработаны также программы SCR22, SCR44, SCR46 и т. д. Их листинги можно заказать по адресу: 261820, Житомирская обл., пгт Ружин, ул. Ленина, 33.

SCR12

```
1 DATA &0012503,&0011505,&0112504,
      &0162704,&0000060,&0006304,
      &0112502,&0162702
2 DATA &0000060,&0062704,&0002400,
      &0077203,&0062704,&0042000,
      &0005303,&0005303
3 DATA &0012737,&0040000,&0000160,
      &0112500,&0104016,&0010400,
      &0012701,&0040000
4 DATA &0012702,&0000012,&0011110,
      &0011160,&0000100,&0005011,
      &0062700,&0000200
5 DATA &0062701,&0000100,&0077211,
      &0062704,&0000002,&0077326,
      &0012737,&0042000
6 DATA &0000160,&0000207
10 FOR I%=&01000 TO &01122 STEP 2
12 READ U%
14 POKE I%,U%
16 NEXT I%
18 DEF USR1=&01000
20 ? CHR$(140);CHR$(140)
22 R$(R)=USR1("10Текст...")
...
```

SCR24

```
0 DATA &0012503,&0011505,&0112504,
      &0162704,&0000060,&0006304,
      &0006304,&0112502
1 DATA &0162702,&0000060,&0062704,
      &0005000,&0077203,&0062704,
      &0042000,&0005303
2 DATA &0005303,&0012737,&0040000,
      &0000160,&0112500,&0104016,
      &0012701,&0040000
3 DATA &0012700,&0000012,&0011146,
      &0042711,&0177400,&0000311,
      &0004767,&0000060
4 DATA &0004767,&0000114,&0062704,
      &0000002,&0012611,&0042711,
      &0000377,&0004767
5 DATA &0000036,&0004767,&0000072,
      &0005011,&0062701,&0000100,
      &0062704,&0000376
6 DATA &0077027,&0162704,&0004774,
      &0077343,&0012737,&0042000,
      &0000160,&0000207
7 DATA &0010446,&0010346,&0005002.
```

```

&0012703, &0000010, &0005004,
&0006111, &0005504
8 DATA &0006102, &0060402, &0006102,
&0060402, &0077310, &0012603,
&0012604, &0000207
9 DATA &0010214, &0010264, &0000100,

```

```

&0010264, &0000200, &0010264,
&0000300, &0000207
10 FOR I7=&01000 TO &01236 STEP 2
...

```

С. МАТВИЙЧУК

Волшебная сила двух ячеек

Вот короткая программа:

```

10 POKE &0600, &012701
20 POKE &0602, &0630
30 POKE &0604, &012503
40 POKE &0606, &011502
50 POKE &0610, &010337
60 POKE &0612, &0122
70 POKE &0614, &0112221
80 POKE &0616, &077302
90 POKE &0620, &012737
100 POKE &0622, &0630
110 POKE &0624, &0124
120 POKE &0626, &0207
130 DEF USR=&0600
140 A$=USR(<строка>)

```

Ее задача — занесение значений в ячейки памяти с адресами 122 и 124, а также копирование параметра в стековую область. Все-го-то! Но дело в том, что ячейки эти не простые: хранящаяся в них информация задает так называемый режим псевдоклавиатуры, когда ввод информации с клавиатуры подменяется на выдачу символов из заданной строки. В ячейку 122 заносится длина этой строки, а в ячейку 124 — ее адрес.

В результате перед программирующими на Бейсике-БК открывается целый ряд дополнительных возможностей:

- создание цепочки программ;
- вывод содержимого экрана на принтер ДВК в КУВТ-86;
- вывод команды CLEAR непосредственно из программы;
- создание всевозможных меню выбора программ для загрузки;
- использование программы MERGE, описанной в выпуске «Вычислительная техника и ее применение» (1990, № 4);
- создание самокомпонующих программ;
- и еще много-много полезного...

Все эти возможности открываются благодаря тому, что после выхода из программы (по команде END или STOP) можно в «автоматическом» режиме исполнить любую директиву Бейсика.

Рассмотрим, например, построение цепочки программ, что позволяет экономить память. При таком методе организации программ сам программный модуль подгру-

жает и запускает модуль, который должен выполняться следующим. Возможен обмен данными между модулями, а также «ручной» ввод данных.

Пусть программа разбита на две части, записанные в файлы с именами P1 и P2. Для работы P2 требуются значения переменных A, B и C, которые формируются в результате работы P1. Тогда программа P1 будет выглядеть так.

```

10 POKE &0600, &012701
20 POKE &0602, &0630
30 POKE &0604, &012503
40 POKE &0606, &011502
50 POKE &0610, &010337
60 POKE &0612, &0122
70 POKE &0614, &0112221
80 POKE &0616, &077302
90 POKE &0620, &012737
100 POKE &0622, &0630
110 POKE &0624, &0124
120 POKE &0626, &0207
130 DEF USR=&0600
140 CR$=CHR$(10) 'код клавиши ВВОД
150 QU$=CHR$(34) 'код кавычки
160 . . .
A=...
. . .
B=...
. . .
C$=...
. . .
A1$="NEW"+CR$+"LOAD"+QU$+"P2"+
QU$+CR$+"RUN"+CR$
A2$=STR$(A)+CR$+STR$(B)+CR$+C$+CR$
A$=USR(A1$+A2$)
END

```

А программа P2 должна содержать оператор

```
INPUT A, B, C$
```

на момент исполнения которого буфер псевдоклавиатуры окажется не пустым (в нем будут значения трех переменных, переданные из P1), и вместо ввода с клавиатуры произойдет считывание из буфера. Программу P2 можно загрузить и вне цепочки (например, для отладки); в этом случае данные для оператора INPUT можно ввести с клавиатуры.

Вывод содержимого экрана на принтер

КУВТ возможен при использовании в качестве программы связи созданной в нашем техникуме программы NT.SAV. Для этого в том месте программы, где нужна печать экрана, нужно вставить команды.

```
A$="BSAVE"+QU$+"TT:GP:"+QU$+
",&D40000,&D100000"
```

```
A$=""+A$+CR$+"CONT"+CR$
A$=USR(A$)
```

Предполагается, что уже исполнены операторы с номерами 10—150 из предыдущего примера.

В. ГУБАРЕНКО

Сумской машиностроительный техникум

Кирпичики ваших программ

Редакционная почта приносит немало писем с просьбами уделять место на страницах журнала небольшим программам и подпрограммам — это лучшая демонстрация возможностей программирования на любом языке. Действительно, это может принести немалую пользу как в аспекте изучения программирования, так и в чисто практическом смысле накопления готовых модулей для использования в своих программах.

В этом номере мы начинаем публиковать такую подборку. Надеемся, вы не останетесь пассивными читателями, а откликнетесь присылкой в редакцию нужных, красивых и нетривиальных текстов подпрограмм.

Постепенно нашими общими усилиями создастся уникальная библиотека подпрограмм и алгоритмов, полезная всем пользователям БК.

Первая программа генерирует экранное меню пяти подпрограмм, каждая из которых демонстрирует тот или иной прием программирования на Бейсике-БК и легко может быть модифицирована под вашу конкретную задачу.

```
10 POKE 136,16384
20 POKE -76,512DR(PEEK(132)/64-40)
30 Y=64
40 CLS
50 LOCATE 0,0,0
60 ? AT(5,5)"Вы можете выбрать:
70 ? AT(5,7)"концентрические круги
80 ? AT(5,9)"сетчатое полотно
90 ? AT(5,11)"хаос прямоугольников
100 ? AT(5,13)"звук вертолета
110 ? AT(5,15)"поздравление
120 ? AT(5,17)"выход
130 Y1=Y
140 LINE (30,Y)-(200,Y+20),1,B
150 C$=INKEY$
160 IFC$=""TH150ELST=ASC(C$)
170 IF T=26THIFY>64THY=Y-20ELSY=164
180 IF T=27THIFY<164THY=Y+20ELSY=64
190 IF T=10THIFY=164TH260ELS220
200 IF Y<>Y1TH LINE(30,Y1)-
(200,Y1+20),0,B
210 GOTO 130
220 LOCATE 5,22,1
230 ON (Y-44)\20 GOSUB 460,270,360,
530,610
240 IF Y=123TH260ELSIFY<>143GOTO 30
```

```
250 IF INKEY$=""TH250ELS130
260 END
270 LOCATE 0,0,0
280 CLS
290 FOR X=0TO30
300 FOR Y=0TO24
310 LOCATE X,Y
320 ? "$";
330 NEXT Y,X
340 IF INKEY$=""TH340
350 RETURN
360 LOCATE 0,0,0
370 CLS
380 FOR B=1TO50
390 X=INT(RND(1)*220)+5
400 Y=INT(RND(1)*280)+5
410 V=INT(RND(1)*40)+20
420 LINE (X,Y)-ST(V,V),,B
430 NEXT B
440 IF INKEY$=""TH440
450 RETURN
460 LOCATE 0,0,0
470 CLS
480 FOR L=8TO15T-1
490 CIRCLE(120,100),L*10+5,,,1,4
500 NEXT L
510 IF INKEY$=""TH510
520 RETURN
530 CLS
540 LOCATE 0,0,0
550 FOR Z=0TO400
560 OUT &D177716,&D100,1
570 OUT &D177716,&D100,0
580 FOR I=0TO50
590 NEXT I,Z
600 RETURN
610 LOCATE 0,0,0
620 CLS
630 ? AT(3,9)"П О З Д Р А В Л Я Е М "
AT(3,9)"* * * * * * * * * * "
AT(3,9)"
640 FOR I=0TO72
650 NEXT I
660 IF INKEY$=""TH630
670 RETURN
```

А вот подпрограмма, составленная В. Роговым и И. Никитиной на ассемблере и позволяющая выводить тексты в служебную строку.

1000	12700	MOV	#14,R0	; ОЧИСТКА
1002	14			
1004	104016	EMT	16	; ЭКРАНА
1006	12700	MOV	#236,R0	; ОЧИСТКА
1010	236			; СЛУЖЕБНОЙ
1012	104016	EMT	-16	; СТРОКИ

Еще три программы прислал в редакцию А. Ланеев. Первая позволяет рисовать с помощью джойстика.

```

1014 12703 MOV #47,R3 ; ДЛИНА
; СТРОКИ
1016 47 ; СИМВОЛОВ
1020 12702 MOV #2000,R2 ; АДРЕС, С
1022 2000 ; КОТОРОГО
; ЗАПИСАНЫ
; СИМВОЛЫ
1024 12701 MOV #0,R1 ; ПОЗИЦИЯ
1026 0 ; ПЕРВОГО
; СИМВОЛА В
; СЛУЖЕБНОЙ
; СТРОКЕ
1030 112200 M:MOV B (R2)+,R0 ; ПЕРЕСЛАТЬ
; В R0 КОД
; СИМВОЛА
1032 104022 EMT 22 ; ФОРМИРОВА-
; НИЕ СИМВОЛА
; В СЛУЖЕБНОЙ
; СТРОКЕ
1034 5201 INC R1 ; УВЕЛИЧЕНИЕ
; НА 1 НОМЕРА
; ПОЗИЦИИ
62 1036 5303 DEC R3 ; УМЕНЬШЕНИЕ
; НА 1 СТРОКИ
; СИМВОЛОВ
1040 3373 BGT M ; ПЕРЕХОД НА
; M, ЕСЛИ
; СТРОКА НЕ
; ЗАВЕРШЕНА
1042 20 RTS ; ВЫХОД

```

Вторая обеспечивает движение по экрану некоей фигуры, пока нажата клавиша.

```

10 X=122
20 Y=122
30 CLS
40 IF PEEK(&0177714)=1 THEN R=-1
50 IF PEEK(&0177714)=3 THEN P=-1
60 IF PEEK(&0177714)=17 THEN P=1
70 IF PEEK(&0177714)=5 THEN R=1
80 PSET(X,Y),1
90 X=X+R
100 Y=Y+P
110 IF X<10 THEN X=240
120 IF X>240 THEN X=10
130 IF Y>230 THEN Y=10
140 IF Y<10 THEN Y=230

```

```

150 PSET(X,Y),2
160 R=0
170 P=0
180 GOTO 40

```

Наконец, третья программа позволяет улучшить оформление текстов на экране. Верхняя часть выведенного ей на экран слова PROGRAM будет зеленой, нижняя — красной.

```

5 COLOR 1,4
10 CLS
20 ? "PROGRAM"
30 FOR P=1 TO 5
40 FOR R=1 TO 58
50 IF POINT(R,P)=1 THEN PSET(R,P),2
60 NEXT R,P

```

При выводе чисел на принтер и экран монитора числа, меньшие 0,01, печатаются в экспоненциальной форме, что неудобно. В вильнюсской версии Бейсика нет специального оператора вывода чисел в форматированном виде, но вместо него В. Кошелев предлагает использовать специальную процедуру, обеспечивающую привычную форму выводимых чисел.

```

10 INPUT B
20 D$=""
30 C$=""
40 T$=""
50 N=VAL(MID$(STR$(B),
; LEN(STR$(B))-1,1))
60 S=INT(CSNG(B)*(10^(N+1))+1/(10^N))
70 T$=STRING$(N-1,48)
80 IF B<.01 THEN 110
90 ? CSNG(B)
100 GOTO 10
110 A$="0."
120 D$=STR$(S)
130 C$=A$+T$+MID$(D$,LEN(D$)-2)
140 ? C$
150 GOTO 10

```

И. ПАНЧЕНКОВ

О символьных переменных

Ошибка вильнюсского интерпретатора Бейсика при выполнении присваивания $A\text{Q} = A\text{Q} + B\text{Q}$ общеизвестна. Для получения правильных результатов в «ИНФО» № 2 за 1990 г., с. 49, предлагались некоторые приемы.

Если приглядеться к работе использующих присваивание $A\text{Q} = A\text{Q} + B\text{Q}$ программ, видно, что при этом «съедается» первый символ, вся информация сдвигается влево, а последний символ дублируется. Ситуация напоминает «дребезг» клавиатуры.

Модифицируем присваивание так, чтобы «съедался» пустой символ: $A\text{Q} = "" + A\text{Q} + B\text{Q}$. Этот прием предлагался консультантом журнала Г. Панченковым, но из-за опечатки он многим оказался непонятен; между тем он обеспечивает правильность не только присваивания, но и выделения подстроки с использованием функций MID\$ и LEN.

Обратим внимание на использование оператора CLEAR при работе с символьными переменными. Буфер символьной переменной рассчитан на 256 символов, следовательно

но, переменная может состоять не более чем из 255 символов. Однако, хотя програма

5 CLEAR 201

работает нормально, стоит лишь заменить 200 на 201, как произойдет сбой. Для его ликвидации придется добавить строку

Спрайты на БК

В «ИНФО» уже рассказывалось о спрайтах на БК-0010 (№ 2, 1990 г.). Подпрограммы, приведенные в той статье, позволяют выводить спрайты на экран лишь в 32 позиции по горизонтали и при отсутствии графического редактора GRED0 практически бесполезны: для записи образа спрайта из Бейсика необходим большой объем двоичных данных, используется зеркальное изображение спрайта, следовательно, есть сложности с цветом...

Решить проблему может специальный редактор спрайтов, точнее, два — для размеров 8×10 и 16×20. Такие редакторы — EDSR8 и EDSR16 — были созданы. Одновременно были разработаны и новые подпрограммы выдачи спрайтов на экран; они позволяют выводить спрайты в 64 позиции по горизонтали и в то же время имеют вдвое меньший объем. Благодаря последнему обстоятельству можно увеличить объем программы на Бейсике, использующей спрайты, или объем блока спрайтов.

При запуске редакторов EDSR сначала требуется ввести восьмеричный номер спрайта, после чего на экране появляются его образ и расширенное изображение. Далее можно использовать команды четырех групп: передвижение курсора; рисование; манипулирование спрайтами; работа с магнитофоном.

Курсор перемещается клавишами управления курсора. При рисовании можно задавать цвет, ставить точку заданным цветом, инвертировать спрайт, закрашивать его текущим цветом, изменять цвет всех точек, чей цвет совпадает с текущим.

Манипулирование спрайтами — это копирование, обмен местами, логические операции И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ. Можно также передвигаться по блоку спрайтов, отменять результаты редактирования, запоминать текущий спрайт.

Файл спрайтов можно записать на МЛ и считать его оттуда.

Подпрограммы выдачи спрайтов на экран помещаются в ОЗУ с адреса 30000, блок

```
10 CLS
20 AX=STRING$(200, "X")
30 ?AX
```

Для максимально полного использования буфера символьной переменной и предотвращения ошибок следует чаще использовать оператор CLEAR для перераспределения памяти под конкретную задачу.

А. ЧЕРНОЛЯСОВ

спрайтов — с адреса 30100. Координата X занесется в ячейку 30074, Y — в ячейку 30076. Номер спрайта (обязательно целое число — % является операндом функции USR, вызывающей подпрограмму. И подпрограммы, и блок спрайтов легко переместить в памяти и использовать обе подпрограммы для выдачи спрайтов на экран.

Ниже приводятся тексты программ, заносимых в ОЗУ подпрограммы выдачи спрайтов.

```
SPR8:
10 DATA 5572,12352,4933,771,26052,20,
      32579,6080
20 DATA 12350,5571,6,3264,32450,26048,
      16384,26560
30 DATA 12348,5569,10,8215,32766,
      -32251,-27376,-27376
40 DATA 26048,62,32328,135
50 FOR I=&030000 TO &030066 STEP 2
60 READ C
70 POKE I,C
80 NEXT
```

```
SPR16:
10 DATA 5572,12352,4933,771,26052,80,
      32579,6080
20 DATA 12350,5571,6,3264,32450,26048,
      16384,26560
30 DATA 12348,5569,20,8215,32764,
      -32249,-27376,-27376
40 DATA -27376,-27376,26048,60,
      32330,135
50 FOR I=&030000 TO &030072 STEP 2
60 READ C
70 POKE I,C
80 NEXT
```

После ввода и выполнения одной из этих программ запускается соответствующий редактор спрайтов и создаются (или загружаются с МЛ) необходимые спрайты, использовать которые в дальнейшем можно, определив подпрограмму выдачи на экран командой

```
DEF USR=&030000
```

Заинтересовавшиеся могут обратиться по адресу: 432012, г. Ульяновск, просп. Гая, 23а, кв. 44; тел. 37-10-57.

П. НИКОНОВ

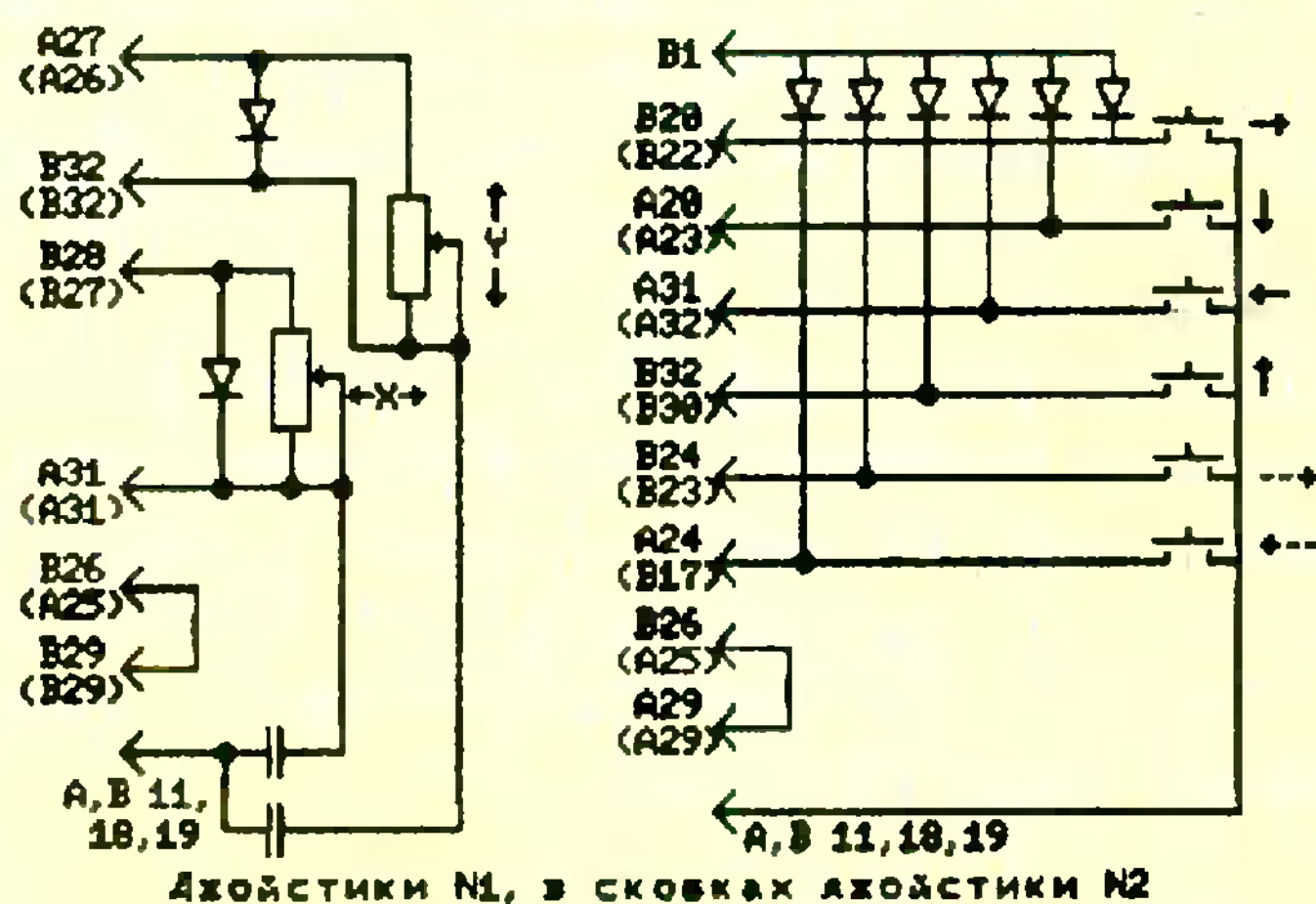
Рекомендации по подключению внешних устройств к порту ввода-вывода БК-0010

64

Номер контакта	Назначение
A1	Сброс
B1	ПРТ
A2	
B2	
A3	
B3	
A4	
B4	
A5	
B5	ВД4
A6	
B6	ВД6
A7	ВД7
B7	ВД5
A8	Питание +5 В
B8	»
A9	»
B9	»
A10	
B10	ВД3; мышь: сброс
A11	Общий
B11	»
A12	ВД2
B12	
A13	ВД1
B13	
A14	
B14	
A15	
B15	
A16	ВД0
B16	
A17	
B17	Мышь: курсор влево. Дискретный джойстик 2: выстрел влево
A18	Общий
B18	»
A19	»
B19	»
A20	КН1 (мышь: клавиша записи). Дискр. джойстик 1: вниз
B20	Дискр. джойстик 1: вправо
A21	
B21	
A22	
B22	КН2 (мышь: клавиша стирания). Дискр. джойстик 2: вправо
A23	Дискр. джойстик 2: вниз
B23	Мышь: курсор вниз. Дискр. джойстик 2: выстрел вправо
A24	Мышь: курсор вправо. Дискр. джойстик 1: выстрел влево
B24	Мышь: курсор вверх. Дискр. джойстик 2: выстрел вправо
A25	ИНД2 (индикация джойстика 2)
B25	

A26	Аналоговый джойстик 2: вертикальный сдвиг
B26	ИНД1 (индикация джойстика 1)
A27	Аналоговый джойстик 1: вертикальный сдвиг
B27	Аналоговый джойстик 2: горизонтальный сдвиг
A28	СТР (принтер: строб)
B28	Аналоговый джойстик 1: горизонтальный сдвиг
A29	ИНДД (индикация дискретного джойстика)
B29	ИНАД (индикация аналогового джойстика)
A30	
B30	Дискр. джойстик 2: сдвиг вверх
A31	Дискр. джойстик 1: влево. Аналоговый джойстик 1: горизонтальный сдвиг
B31	ЗП (принтер: запрос приемника)
A32	Дискр. джойстик 2: влево
B32	Дискр. джойстик 1: вверх. Аналоговый джойстик 1: вертикальный сдвиг

Предлагаемый стандарт предусматривает возможность одновременного подключения печатающего устройства (принтера), устройства ввода УВК-01 («мышь») и двух джойстиков (аналоговых или дискретных). Схемы



джойстиков приведены на рисунке, а опознавание их типов происходит автоматически по наличию контрольных перемычек между контактами B26 и A29 или B29 (для джойстика 1) и контактами A25 и A29 или B29 (для джойстика 2). Например, наличие перемычек B26 — B29 и A25 — A29 означает, что подключен джойстик 1 аналогового типа и джойстик 2 дискретного типа, наличие перемычки A25 — B29 означает, что подключен только джойстик 2 аналогового типа.

Для подключения принтера используются девять выходных регистров (ВД0 — ВД7 данные и ВД8 строб) и один входной (ВВ8 ЗП). Остальные запросы (подразумевается, что они есть) заземляются на контакты А, В (11, 18, 19).

Важные восьмеричные числа

При программировании на ассемблере возникает необходимость помещения кода КОИ-7 в младший или старший байт регистра общего назначения. Если в случае младшего байта никаких проблем нет (например, MOV # 12, R2), то при занесении кода в старший байт необходимо сначала подобрать соответствующее восьмеричное число. Допустим, мы используем команду EMT 10 и нам необходимо занести в старший байт регистра R2 в качестве ограничителя строки код 12 (ВВОД). Для этого необходимо поместить в R2 восьмеричное число 5000: MOV # 5000, R2. В результате в младшем байте R2 окажется 0, а в старшем — 12. Для этого случая в ассемблере МИКРО10 есть специальная псевдокоманда .В:, но как вычислить физический адрес метки? К тому же такие команды есть не во всех версиях ассемблеров.

Поэтому в некоторых случаях может оказаться полезным непосредственное занесение в регистры восьмеричных чисел, соответствующих нужным кодам. Такие числа для некоторых управляющих кодов приведены в таблице. Из нее легко усматривается законо-

мерность, облегчающая поиск соответствующих чисел для любых других кодов.

Восьмеричные		Назначение
код	число	
10	4000	—
12	5000	ВВОД
14	6000	СЕР
15	6400	УСТ ТАБ
16	7000	РУС
17	7400	ЛАТ
20	10000	СЕР ТАБ
22	11000	┌
23	11400	ВС
24	12000	ГТ
26	13000	←
27	13400	→
30	14000	←+
31	14400	→+
32	15000	↑
33	15400	↓
34	16000	↖
35	16400	↗
36	17000	↘
37	17400	↙

В. КИМ

ДОСКА ОБЪЯВЛЕНИЙ

Обмениваюсь программами для БК. Имею около 500 различных программ.

220088, Минск, ул. Антоновская, 32, кв. 8. М. М. Погореляк.

Хочу переписываться и обмениваться программами с владельцами БК-0010-01. Имею более 500 программ в кодах. Интересуюсь программированием в кодах.

399113, Липецкая обл., Липецкий р-он, с. Частая дубрава, п. Московский, 7, кв. 2. И. А. Кондрашов.

Предлагаю обмен программами (в кодах, на Бейсике, Т-языке, Фокале) для БК-0010. Имею более 200 программ.

150023, Ярославль, ул. Рыкачева, 4/7, кв. 18. А. В. Новиков.

Хотел бы обмениваться программами и опытом работы на БК-0010. Имею более 200 программ.

490050, Семипалатинск, ул. Урицкого, 11, кв. 12. А. П. Рязанов.

Наш клуб «ИНФО» стремится помочь в общении всем единомышленникам в области информатики. Мы работаем на многих типах ЭВМ, но основные — это БК-0010 и IBM

РС. Наш адрес: 422520, Татария, Зеленодольск-10, а/я 161, клуб «ИНФО». С. В. Фомину.

Хотел бы переписываться с владельцами БК-0010. Имею 400 программ. Готов поменяться и поделиться. Высылаю каталог.

357359, Ставропольский край, Предгорный р-н, Юца-1, ул. Овражная, 22. Д. Б. Гладько.

* * *

Сегодня мы публикуем также несколько адресов клубов программистов, в которых занимаются, в частности, и компьютерами БК-0010.

658710, Алтайский край, Камень-на-Оби, ул. Пушкина, 37, кв. 30. М. Б. Бурлуцкому.

420015, Казань, а/я 4427. Объединение профессиональных программистов. А. М. Кучерову.

198013, Ленинград, а/я 65. Клуб «Компьютер» Ленинградского технологического института им. Ленсовета. А. Г. Ефимову.

НАША МАРКА — «ИМПАКТ»

Научно-техническое малое предприятие «Импакт» — преемник объединения «Импульс», которое одним из первых в стране взяло на себя обслуживание пользователей БК-0010 (-01) и получило широкую известность благодаря наиболее полному и систематизированному фонду программного обеспечения для этих компьютеров.

«Импакт» поставляет индивидуальным пользователям и организациям программное обеспечение для компьютеров БК-0010 (-01).

Программы поставляются в комплектах, среди которых особо популярны «25 и 50 лучших игр в кодах», «Основной комплект», «Минимакс» (наиболее качественные программы для БК). Раз в два месяца выпускаются комплекты «Лучшие новинки». Поставляются также отдельные программы.

Стоимость программы в комплекте — 4 рубля. В отличие от других организаций «Импакт» не только поставляет программы, но и предоставляет бесплатные консультации заказчикам.

Организацией разработан «Каталог-справочник программ для БК-0010 (-01)», который, в отличие от обычных списков программ, является не рекламой, а справочником, постоянно необходимым пользователю БК. Каталог включает краткое описание и данные о 1300 программах. Цена — 17 рублей с пересылкой, для предприятий и организаций — 49 рублей.

«Импакт» — первая и пока единственная организация в стране, поставляющая программы для компьютера БК-11.

С авторами программ заключаются договоры на тиражирование.

Детальный проспект об услугах высылается бесплатно.

Не забудьте пометить в своей записной книжке наш адрес:

252124, Киев-124, а/я 8, «Импакт».

Тел.: 444-45-50, 440-91-20, 432-17-61; факс: 227-08-37 (код 0-44).

«Импакт»: три года работы в условиях рынка — гарантия успеха.

ЖУРНАЛ «БЮЛЛЕТЕНЬ БК»

С января 1990 г. «Импульс» издает ежемесячный информационно-справочный журнал для пользователей компьютера БК. Он зарегистрирован как официальное всесоюзное издание.

В каждом номере журнала публикуются подробное описание до 100 новых программ, информация о малоизвестных программах, инструкции, листинги программ, описания доработок БК, статьи о «секретах» программирования, ответы на вопросы читателей.

Начата публикация учебника по программированию в кодах на БК для начинающих, который, в отличие от всех других, построен исключительно на практических материалах. Печатается и полное техническое описание БК, которое ранее нигде не публиковалось.

Для пользователей УКНЦ и БК-11 помещается бесплатное приложение.

Стоимость номера с пересылкой — 6 руб. 40 коп., годовая подписка — 64 руб. (скидка 15 %). С 1992 г. журнал занесен в каталог «Союзпечати», в 1991 г. его можно заказать наложенным платежом. Условия подписки заказывайте по адресу: 252142, Киев-142, а/я 14, редакция журнала «Бюллетень БК». Тел.: 444-45-50.

В. ПРОХОРОВ, С. РАКОВ, В. КИРЮХИН

Олимпиадные задачи по информатике

Рассмотрим задачу (автор В. Прохоров), предлагавшуюся на практическом туре III Всесоюзной олимпиады по информатике в г. Харькове в 1990 г. Напомним ее условие.

$$\text{Обозначим } S(n, a, b) = \sum_{k=1}^n \frac{k}{[\log_b k]},$$

где $\sum_{k=1}^n x_k = x_1 + x_2 + \dots + x_n$; $[\log_b k]$ означает целую часть $\log_b k$; через $\log_b k$ обозначено такое число z , что $b^z = k$.

Требуется составить программу, которая выводит результат с возможно большим числом верных десятичных знаков для трех случаев:

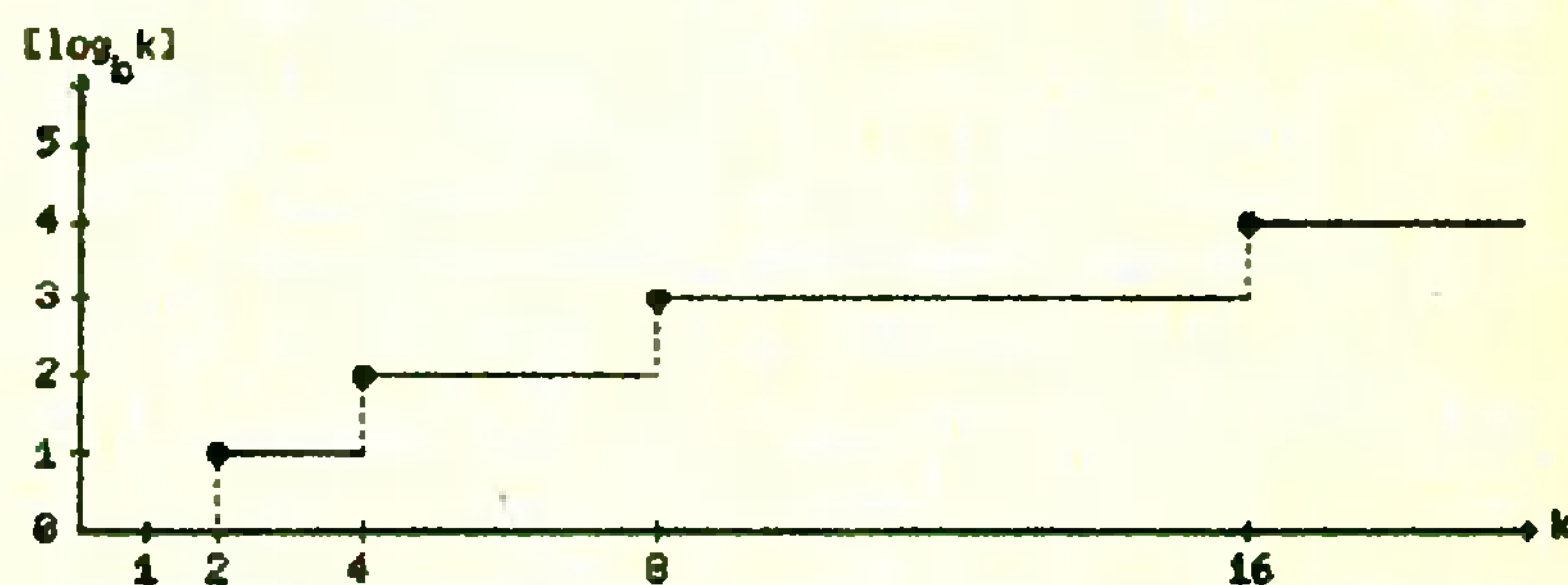
1. $S(100, 3, 2)$;
2. $S(100, 3, 1.9)$;
3. $S(100, 3.1, 1.9)$.

На первый взгляд задача кажется достаточно простой. Однако это обманчивое впечатление. Для ее правильного решения необходимо довольно хорошо знать, каким образом происходит обработка информации на ЭВМ. Не правы те, кто полагает, что приближенность решения вычислительных задач — это органически присущее свойство любого компьютера. Наоборот, для многих вычислительных задач можно получить абсолютно точный ответ, настолько точный, насколько это позволяют сделать исходные данные. Представленная задача хорошо иллюстрирует этот факт, и на ее примере можно получить ответ, как найти решение без какой-либо погрешности, т. е. абсолютно точно, несмотря на специфику обработки информации на ЭВМ.

Идея решения задачи

Если внимательно проанализиро-

вать, что представляет собой искомая сумма $s(n, a, b)$, то можно увидеть следующее. Показатель степени $[\log_b k]$ в зависимости от k является кусочно-постоянной функцией. Например, для $b=2$ эта функция представлена на рисунке.



Легко видеть, что при изменении величины K от 1 до 100 знаменатель увеличится не более чем в 7 раз. Поэтому для представления знаменателя достаточно всего 11 десятичных разрядов ($10^{10} < 31^7 < 10^{11}$).

Далее, для вычисления знаменателя не следует использовать стандартные функции вычисления логарифма и целой части числа — это может привести к потере точности и не эффективно. Лучше накапливать умножением величину b^p , присвоив ей в начальный момент значение b и затем умножая на b каждый раз, когда текущее значение K удовлетворит неравенству $K \geq b^p$ (т. е. использовать величину b^p в качестве индикатора изменения знаменателя: в момент изменения индикатора необходимо и сам знаменатель изменить, умножив его на a). Еще раз обратим внимание на то, что величины b^p и $[\log_b k]$ могут быть вычислены точно, если количество разрядов для представления этих чисел будет не меньше 11 (таким образом, для их хранения можно использовать числовой тип «действительный» в Паскале или «двойной точности» в Бейсике).

Суть идеи решения поставленной задачи заключается в представлении данных для хранения текущих значений суммы и слагаемого в виде пары действительных чисел (количества разрядов типа «целый» недостаточно) и выполнении над ними только арифметических операций сложения (согласно правилам сложения дробей) и сокращения дробей. Следуя этим путем, можно получить результат в виде обыкновенной дроби без погрешностей. После этого остается выводить последовательно знаки десятичного представления результата. Для выделения периода получающейся десятичной дроби целесообразно воспользоваться следующей известной теоремой.

68 **Теорема.** Для того чтобы правильная несократимая дробь представлялась чистой периодической дробью, необходимо и достаточно, чтобы ее знаменатель не делился нацело на 2 и 5.

Использовать эту теорему можно следующим образом. Началу периода будет соответствовать тот момент разложения обыкновенной дроби в десятичную, когда у оставшейся неразложенной части знаменатель не будет делиться нацело ни на 2, ни на 5. Концу периода — момент, когда остаток от деления совпадает с остатком, соответствующим началу периода.

И последнее замечание. Для сокращения дробей приходится отыскивать наибольший общий делитель числителя и знаменателя, представленный в виде действительных чисел (предварительно числитель и знаменатель необходимо умножить на степень 10, для того чтобы они стали натуральными). Для этого можно воспользоваться классическим рекуррентным алгоритмом Евклида, вычитая из большего числа меньшее до тех пор, пока остаток не станет равным нулю (тогда НОД будет равен первой отличной от нуля разности).

Для ускорения алгоритма Евклида его можно модифицировать, используя вместо разности остаток от деления большего числа на меньшее. Единственное затруднение будет состоять в том, что в некоторых реализациях языков программирования нет действительно значных встроенных функций, вычис-

ляющих целую часть действительного числа (функция *trunc(x)* имеет своим значением число типа «целое»). Если это так, то для выполнения необходимо разработать некоторую вспомогательную функцию.

Программа реализации алгоритма

Для программной реализации мы выбрали алгоритмический язык Паскаль из-за удобных изобразительных средств и средств структуризации, позволяющих достаточно просто разобраться в тексте самой программы. Без дополнительных переделок представленная ниже программа может быть выполнена в системе *TURBO PASCAL v.5.0* на *IBM PC*. Для ПЭВМ «Ямаха» с системой *TURBO PASCAL v.3.0* требуется внести некоторые коррективы, в частности реализовать свою процедуру-функцию *inf(x)* с учетом имеющихся там возможностей.

В обобщенном виде программа, реализующая вышеназванную идею и имеющая имя *Sum*, представляет собой последовательность выполнения ряда процедур для каждого из трех наборов параметров *n, a, b*. Следует отметить, что в задаче не требовалось получить решение для любых названных параметров. Поэтому в данном случае стремление к всеобщности является излишним. Более того, учет конкретных значений указанных параметров позволяет в определенном смысле упростить искомое решение.

Основными процедурами в программе *Sum* являются процедура вычислений (*Calculations*) и процедура вывода результатов в виде обыкновенной десятичной и десятичной периодической дроби (*Output Result*). Поскольку текст программы снабжен подробными комментариями, то читатель без труда сможет сам разобраться в сути выполняемых этой программой действий.

```

procedure OutputResult (p,q:real);
{=====}
{      вывод результатов расчетов      }
{=====}
procedure OutputVulgarFraction;
{=====}
{      вывод результата в виде обыкновенной дроби      }
{=====}
begin
  writeln('Точное значение в виде обыкновенной дроби:');
  writeln('S=',p,'/',q);
end ;{OutputVulgarFraction}
procedure OutputDecimalPeriodicfraction;
{=====}
{      вывод результата в виде десятичной      }
{      периодической дроби      }
{=====}

```

```

var
  Rest : real;
  BeginningPeriodRest : real;
  TempDigit : integer;
  PeriodPartFlag : boolean;
  r : integer;
procedure BeginPeriodSearch;
begin
  CancellationFraction (Rest, q);
  if not Devide (2,5,q)
  then
    begin
      PeriodPartFlag :=TRUE;
      BeginningPeriodRest :=Rest;
      write ('(');
    end { if - then }
end; {BeginPeriodSearch}
function EndPeriod : boolean;
begin
  { логическая функция, принимающая значения }
  { TRUE - в конце периода дроби и }
  { False - в противном случае }
  EndPeriod:=(Rest = BeginningPeriodRest);
end; {EndPeriod}
begin
  write ('Точное значение суммы в виде');
  writeln('десятичной периодической дроби:');
  r:=trunc (p/q);
  write (r, ' ');
  PeriodPartFlag := FALSE;
  Rest:=(p-r*q);
  repeat
    if not PeriodPartFlag
    then
      begin
        BeginPeriodSearch;
        TempDigit:=trunc (Rest*10/q);
        write (TempDigit);
        Rest:=(Rest*10 - TempDigit*q);
      end
    until EndPeriod;
  write (' ');
end; {OutputDecimalPeriodicFraction}
begin
  OutputVulgarFraction;
  OutputDecimalPeriodicFraction;
end; {OutputResult}
for Variant:=1 to 3 do
  begin
    writeln;writeln;writeln;
    write('S(100',Array[Variant]);
    writeln ('.',Barray[Variant],')');
    Calculations;
    OutputResult (Numerator, Denominator);
  end;
end; { Sum }

program Sum;
const
  N = 100;
  Aarray: array[1..3] of real=(3, 3, 3.1);
  Barray: array[1..3] of real=(2, 1.9, 1.9);
var
  Numerator : real;
  Denominator : real;
  Variant : byte;
function Nod (p,q : real) : real;
begin
  if(p<>0) and (q<>0)
  then
    if p>=q
    then
      Nod:= Nod(q, p - int(p/q)*q)
    else
      Nod:=Nod(q,p)
    else
      if q=0
      then
        Nod:=p
      else
        Nod:=q
  end; {Nod}
procedure CancellationFraction (var p, q : real);
var
  Nodpq :real;
begin
  Nodpq:=Nod (p,q);
  p:=p/Nodpq;
  q:=q/Nodpq;
end; {CancellationFraction}
function Devide (a, b, c : real) : boolean;
begin
  Devide:=(c=int(c/a)*a) or (c= int(c/b)*b);
end; {Devide}
procedure Calculations;
var
  A, B : real;
  k : integer;
  DemonChangeIndicator : real;
begin
  A:=Aarray [Variant];
  B:=Barray [Variant];
  DemonChangeIndicator:=B;
  Numerator :=0;
  Denominator :=1;
  for k := 1 to N do
    if k>=DemonChangeIndicator
    then
      begin
        DemonChangeIndicator:=DemonChangeIndicator*B;
        Denominator :=Denominator*A;

```

```

Numerator:=Numerator*K;
end
else
  Numerator:=Numerator-K;
while (Denominator-int(Denominator))>0 do
  begin
    Numerator:=Numerator*10;
    Denominator :=Denominator*10;
  end;
CancellationFraction(Numerator, Denominator)
end; {Calculations}

```

Анализ результатов выполнения программы

Как уже было сказано, представленная программа позволяет получить абсолютно точное значение суммы $S(n, a, b)$ для трех указанных наборов данных.

$S(100, 3.0000000000E+00, 2.0000000000E+00)$

Точное значение в виде обыкновенной дроби:

$S = 1.7188000000E+04 / 7.2900000000E+02$

Точное значение суммы в виде десятичной

периодической дроби:

23.(577503429355281207133058984910836762688614540466392318244170096021947873799725651)

$S(100, 3.0000000000E+00, 1.9000000000E+00)$

Точное значение в виде обыкновенной дроби:

$S = 3.7918000000E+04 / 2.1870000000E+03$

Точное значение суммы в виде десятичной

периодической дроби:

17.(337905807041609510745313214449016918152720621856424325560128029263831732967535436671239140374942844078646547782350251486053955189757658893461362597165066300868770004572473708276177411979881115683584819387288523090992226794695930498399634202103)

$S(100, 3.1000000000E+00, 1.9000000000E+00)$

Точное значение в виде обыкновенной дроби:

$S = 4.2094898466E+12 / 2.751261411E+11$

Точное значение суммы в виде десятичной

периодической дроби:

15.(30021767340485726492881780962033882259710008429275824786385884531258305095971855890503408146688014383570997933409801115282441593457230767577579289240676439590840737176297766714745474010)

Анализируя полученные результаты, можно сказать следующее.

Результаты вычислений $S(100, 3, 2)$ и $S(100, 3, 1.9)$ в пояснениях не нуждаются. Обратимся к $S(100, 3.1, 1.9)$. Знаменатель результата, представленного в виде обыкновенной дроби, вычислен точно (содержит все 11 разрядов), а для представления числителя не хватило двух разрядов (число оказалось 13-значным). Как выйти из этого положения? На *IBM PC* при наличии арифметического сопроцессора можно воспользоваться вместо типа «действительный» типами «двойной» или «повышенной точности». При отсутствии сопроцессора на *IBM PC* или на «Ямахе» преодолеть возникающие трудности сложнее. Один из возможных подходов

состоит в моделировании типа «длинное целое», которое допускает представление по крайней мере 13-разрядных целых чисел, с последующим представлением рациональных чисел в виде пары чисел типа «длинное целое». Кроме того, использование Бейсика (MSX) существенно упрощает проблему, так как числа с двойной точностью там имеют 14 знаков в мантиссе.

К сожалению, никто из участников олимпиады не справился с задачей в полном объеме, т. е. не нашел все верные десятичные знаки суммы $S(n, a, b)$. Жюри при оценке решений учитывало количество верных знаков в ответе. При равном количестве верных знаков в двух работах лучшей считалась та, в которой результат достигался за наименьшее время.

70 Следует выделить следующие подходы, используемые участниками олимпиады при решении поставленной задачи:

вычисление результата «в лоб» в соответствии с принципом «что первое пришло в голову, то и делаю»;

моделирование «длинной арифметики»;

вычисление с помощью операций над обыкновенными дробями.

Ни один из участников олимпиады не приблизился к мысли о периодичности получающейся обыкновенной десятичной дроби. Из тех, кто был наиболее близок к требуемому решению, можно выделить О. Таборовца (89 баллов из 100), А. Демакова (85 баллов), А. Челышева (80 баллов) и Р. Варблана (75 баллов).

Другие возможные подходы к решению задачи

Мы рассмотрели решение представленной задачи, наиболее доступное многим школьникам. Однако привлекательность задачи состоит в том, что она предполагает и другие подходы к ее решению, позволяющие открывать новые грани процесса обработки информации на ЭВМ. Мы надеемся, что интересующийся читатель сам сможет довести опи-

санные ниже идеи до логического конца, и мы не сомневаемся в той пользе, которую он при этом получит.

Первый подход, с которого мы начнем, основан на использовании системы счисления с подходящим основанием p . Нетрудно заметить, что все числа, используемые в вариантах 1 и 2 условия задачи, представляются конечными дробями в системе счисления с основанием $p=3$. Аналогично и в варианте 3 все числа представляются конечными дробями в системе счисления с основанием $p=31$. Причем для их хранения достаточно иметь 7 разрядов после запятой (впрочем, можно обойтись и только целыми числами с соответствующим числом разрядов). С учетом этого можно реализовать арифметику в p -ричной системе счисления, а затем полученный точный результат обратить в десятичную периодическую дробь по уже рассмотренной схеме.

Второй подход, о котором хотелось бы упомянуть, связан с арифметикой периодических дробей. Наряду с традиционной можно реализовать арифметику, использующую представление чисел в виде периодических дробей. Для случаев 1 и 2 это вполне реально, так как наибольший возникающий при этом период не будет превышать 729 разрядов. Для случая 3 длина периода, как легко заметить, больше.

В заключение отметим, что вычисления с высокой степенью точности, так же как и вычисления без погрешностей, являются важной составляющей раздела прикладной информатики, называемого компьютерной алгеброй. Современные пакеты поддержки математической деятельности типа *DERIVE* позволяют получить точный результат «в лоб», запросив точное значение суммы в виде обыкновенной дроби или десятичной периодической дроби.

И последнее, величину $S(n, a, b)$ можно вычислить аналитически, используя при этом арифметические операции и операцию возведения в степень. Мы оставляем это читателю для самостоятельного рассмотрения.

Случай в городе Глупове

Отцы г. Глупова решили для увековечения юбилея города соорудить на центральной площади напротив городской думы памятный монумент.

Местные монументалисты изваяли памятник в виде строгого гранитного куба, на одной из сторон которого высекли «г. Глуповъ», год его основания и горельеф первого градоначальника. Для установки памятника из столицы была приглашена бригада кооператоров-монтажников, которая досрочно доставила монумент и уложила его на указанное место площади.

Только во время городского митинга, когда покрывало под звуки оркестра упало с монумента, городские руководители и простые обыватели с ужасом увидели, что хотя памятник и был установлен в точности на требуемое место, но совершенно неприличным образом... При установке не была обеспечена его нужная ориентация — надпись и физиономия первого градоначальника должны были быть обращены к думе, притом «вниз ногами».

Ситуация казалась безвыходной: ни поднять, ни сдвинуть глыбу вручную было невозможно, единственный глуповский подъемник тяжестей был давно сломан, на повторный вызов столичных кооператоров в городской казне уже не хватало денег...

Но тут один местный изобретатель заявил, что он придумал, как преодолеть возникший конфуз. Ведь хотя вручную гранитное творение и не сдвинуть с места, но при известном энтузиазме его вполне возможно кантовать (перекатывать через ребра) «всем миром». А изобретатель нутром чуял, что, осуществив последовательность надлежащим образом выбранных кантований, можно поместить монумент в точности туда же, где он стоял перед этим, но в любой другой требуемой ориентации. Благо, что памятник был установлен прямо на мостовую без всякого постаментов и его перекачиванию ничто в окрестности не мешало. Заметим, что место установки памятника было выбрано не случайно и менять его никоим образом не позволялось.

Как же для известного начального положения памятника определить последовательность кантований, после выполнения которой он будет лежать на прежнем месте, но уже в необходимой ориентации?

Решение. Опишем прежде всего задачу на математическом языке. Будем полагать, что памятник — идеальный куб, площадь — идеальная «бесконечно шершавая» (т. е. проскальзывание куба исключается) плоскость. Перенумеруем грани таким образом, что если памятник поставить надписью к нам и «вниз ногами», то грань с надписью будет иметь номер 0, верхняя — номер 1, правая — 2, а остальные грани — номера на 3 больше, чем им противоположные.

Сопоставим каждой ориентации куба пару чисел (i, j) , где i — номер грани, обращенной в сторону думы (в понятном смысле, хотя формулировка не вполне точная), а j — номер верхней грани.

Как и при рассмотрении любой проблемы, проанализируем, всегда ли решение существует (т. е. все ли сочетания исходных данных допустимы). Кроме всего прочего, отбрасывание исходных данных, для которых задача неразрешима, может позволить упростить решение (в крайнем случае даже может оказаться, что задача вообще не имеет решения, за исключением тривиальных случаев). Кто-то скажет: зачем делать лишнюю работу? Ведь в условии задачи достаточно ясно говорится, что требуется найти решение, значит, оно существует. Но мало ли что говорится в условии! Ведь и мудрецы могут ошибаться...

Итак, введем на каждой грани куба «ось ориентации» («ось» грани) таким образом, чтобы на всякой паре смежных (имеющих общее ребро) граней они были бы взаимно перпендикулярны. Это действительно можно сделать. Например, ось ориентации грани 0 будем полагать перпендикулярной грани 1, для грани 1 — грани 2, для грани 2 — грани 0. Для остальных граней оси ориентации введем из имеющего место правила: для параллельных (противоположных) граней оси ориентации параллельны.

При одном кантовании куба внизу оказывается грань, смежная той, которая была внизу перед этим (смежная по тому ребру, через которое проводилось кантование). Отсюда следует, что при каждом кантовании направление оси текущей нижней грани будет меняться на перпендикулярное прежнему. Таким образом, поскольку для возвращения куба на прежнее место требуется четное число кантований (докажите!), оси граней, находящихся внизу до и после последовательности кантований, будут совпадать.

Таким образом, все возможные ориентации куба можно разбить на два класса по направлению оси ориентации нижней грани, причем если начальная и конечная ориентации куба принадлежат разным классам, то никакой последовательностью кантований задачу решить нельзя (разумеется, это не означает, что если начальная и требуемая конечная ориентации принадлежат одному классу в этом смысле, то задача разрешима; мы пока получили лишь достаточное условие неразрешимости задачи, или необходимое разрешимости).

Заметим, что при введенной нами нумерации ориентаций куба класс ориентации начального положения определить достаточно просто. Для этого учтем, что ось обращенной к нам грани будет вертикальной, в частности, при ориентациях куба $(0, 1)$, $(1, 2)$, $(2, 3)$ (это следует непосредственно из введенного нами выше правила ориентации осей граней). Для перечисленных трех ориентаций куба справедливо условие $j-i=1$. Кроме указанных вертикальную ось передней грани будут иметь ориентации, отличающиеся от перечисленных заменой любой из граней в паре (i, j) (или обеих) на противоположную, и только они! (Докажите.) Отсюда следует, что класс, которому принадлежит ориентация (i, j) , можно определить по значению характеристики $(j-i) \bmod 3$. Если она равна 1, то это один класс ориентации, а если 2 — другой (иных значений эта характеристика для данной нумерации граней куба не принимает).

Поскольку последовательности из нуля и двух кантований не меняют ориентации куба, рассмотрим простей-

ший нетривиальный случай — из четырех кантований, причем исключим такие четверки, где все кантования «вперед» — «назад» либо «вправо» — «влево» (они тоже не меняют ориентации). Более того, исключим даже те последовательности, где подряд идут действия «вперед» — «назад» или «вправо» — «влево», поскольку и такие пары не меняют ориентации куба.

Для возврата куба на прежнее место необходимо, чтобы число кантований «вправо» совпадало с числом кантований «влево», а число «вперед» — с числом «назад». Таким образом, в случае четырех кантований можно рассматривать лишь случаи, когда последовательность состоит из каким-то образом упорядоченных кантований «вперед», «назад», «направо», «налево» (ровно по одному разу). С учетом сказанного ранее каждая такая последовательность определяется первыми двумя элементами (8, а не 16 вариантов! Почему?), а остальные два определяются через них. Назовем такие четверки примитивными.

Нетрудно убедиться, что при реализации последовательности примитивных кантований одна из вершин куба остается неподвижной, а куб обкатывает ее, оказываясь в конце лежащим на том же месте, что и вначале. Таким образом, в этом случае фактически происходит как бы поворот куба относительно одной из его главных диагоналей на треть оборота (к вершине куба прилежат три грани, а при атомарных кантованиях куб последовательно лежит на четырех из них!).

При любом повороте куба относительно любой главной диагонали на треть оборота передней гранью оказывается одна из граней, примыкавших к передней до поворота, причем выбором диагонали и направления поворота можно обеспечить любую из двух возможных ориентаций куба для любой из возможных при этом передних граней (покажите!). Перебирая все восемь возможных примитивных четверок кантований (когда центр куба описывает каждую из четырех возможных траекторий в каждом из двух возможных направлений), мы получаем все восемь возможных поворотов куба на треть оборота вокруг глав-

ной диагонали (четыре диагонали по два направления поворота для каждой).

Таким образом, примитивные четверки кантований позволяют решить задачу, если начальная и конечная ориентации куба относятся к одному классу и если грань, которая должна стать фронтальной, вначале примыкает к передней. Конкретную последовательность, решающую задачу, в этом случае проще всего определить прямым перебором восьми вариантов (тут-то нам и пригодится ЭВМ).

Следовательно, примитивные четверки кантований не позволяют решать задачу лишь в том случае, когда грань, которая должна стать передней, вначале является задней или передней (в последний случай попадает и тривиальный вариант, когда требуемая конечная ориентация совпадает с начальной, хотя есть и не тривиальный, когда надпись обращена в требуемую сторону, но «вверх ногами»). Однако применением произвольной примитивной четверки кантований как заднюю, так и переднюю грани куба можно сделать «не задней» и «не передней». Значит, применением композиции из пары последовательно примененных примитивных четверок кантований (где первая — любая примитивная) можно решить задачу, если только начальная и конечная ориентации куба принадлежат одному классу.

Итак, мы показали следующее.

1. Условие $(i0-j0) \bmod 3 = (ik-jk) \bmod 3$ есть необходимое и достаточное условие разрешимости задачи.

2. Если задача разрешима и грань, которая должна стать фронтальной, не является вначале ни передней, ни задней, то задача решается применением одной из восьми примитивных четверок кантований.

3. Если грань, которая должна стать фронтальной, является вначале задней или передней (в последнем случае разумно отдельно выделить ситуацию, когда куб с самого начала лежит в требуемой ориентации и с ним можно ничего не делать), то задача может быть решена последовательным применением двух атомарных четверок кантований, где применение сначала произвольной атомарной четверки кантований сводит

проблему к ситуации, рассмотренной в п. 2. (А можно ли в этом случае обойтись, хотя бы и не всегда, меньшим числом кантований, а именно шестью?)

Дадим описание алгоритма на языке Альфа (разработанном в 1960 г. в Новосибирске группой под руководством А. П. Ершова). Трансляторы с Альфы (разработанные той же группой) имелись на различных советских ЭВМ, начиная с М-20 (была разработана в ИГМиВТ АН СССР в 1958 г.; объем ее ОЗУ на ферритовых сердечниках составлял всего около 20К байт, хотя имелась внешняя память на магнитных барабанах — около 60К байт — и на магнитной ленте — около 1,5М байт; М-20 была собрана на 4,5 тыс. электронных ламп и 35 тыс. полупроводниковых диодов), затем М-220, М-222, БЭСМ-6 и наконец «Эльбрус».

Заметим к слову, что система Альфа, кроме всего прочего, обладает очень важным достоинством — она позволяет получать после трансляции программы весьма высокой эффективности по объему и быстродействию. Это благодаря использованию очень качественной автоматической оптимизации пользовательской программы во время ее компиляции.

Для более компактной записи мы используем сокращенные представления ключевых слов. Заметим к слову, что Альфа допускает любые их сокращения, пока не теряется однозначность понимания, притом как русские, так и английские; более того, Альфа понимает и иероглифы — когда одной кнопке соответствует сразу целое понятие языка без всяких слов. Использовали когда-то (лет 30 назад) такое устройство ввода программы, на клавиатуре которого кроме кнопок с русскими, латинскими и греческими буквами и цифрами были кнопки с надписями «Начало», «Процедура» и т. п. (каждой из которых соответствовал свой код).

Надеемся, что вы разберетесь в этой программе, даже не зная языка Альфа.

Программа Кубик;

Прим-----;

Нач

Цел Масс $m[\emptyset:5, \emptyset:2]=$

1 0, 1, 2 1, 2, 1, 3 1, 2, 4, 0 1,
3, 1, 0 1, 3, 4, 2 1, 0, 4, 5 1,
3, 1, 5 1, 3, 4, 3 1;

Примеч: тройки номеров граней,
примыкающих к одной вершине,
в порядке обхода по часовой
стрелке;

Прим-----;
неразрешимо(i_0, j_0, ik, jk)=
 $((i_0 - j_0) \bmod 3 < (ik - jk) \bmod 3)$;

Примеч: логическая функция,
определяющая, принадлежат ли
 (i_0, j_0) и (ik, jk) к одному
классу;

Прим-----;
Цел Проц правая(i, j);

Примеч: определение для позиции
 (i, j) номера правой грани по
таблице m ;

Нач Цел k, l

Для $k := 0, \dots, 5$ Вып

Для $l := 0, 1, 2$ Вып

Если $i = m[k, l]$ И

$j = m[k, (l+1) \bmod 3]$

То На выход;

вывод($0, 'т', 'Ошибка в обращении к$
"правая":', '\

' грани i и j смежные!');

выход: правая: $= m[k, (l+2) \bmod 3]$

Кон;

74 Прим-----;
левая(i, j) $= (правая(i, j) + 3) \bmod 6$;

Примеч: определение для позиции
 (i, j) номера левой грани по
таблице m ;

Прим-----;
задняя(i, j) $= (i + 3) \bmod 6$;

Примеч: определение для позиции
 (i, j) номера задней грани куба;

Прим-----;
нижняя(i, j) $= (j + 3) \bmod 6$;

Примеч: определение для позиции
 (i, j) номера нижней грани куба;

Прим-----;
Проц поворотик($i_0, j_0, r, i1, j1$);

Аргументы i_0, j_0, r ;

Результаты $i1, j1$;

Примеч: поворот одним кантованием,
 $r = 0, \dots, 3$ (вправо, вперед,
влево, назад соответственно);

Нач Перекл выбор[$0:3$] = вправо,
вперед, влево, назад;

На выбор[r];

вправо: $i1 := i$;

$j1 :=$ левая(i, j); На все;

вперед: $i1 :=$ нижняя(i, j);

$j1 := i$; На все;

влево: $i1 := i$;

$j1 :=$ правая(i, j); На все;

назад: $i1 := j$;

$j :=$ задняя(i, j); На все;

все:

Кон;

Прим-----;
Проц поворотN(i_0, j_0, N, k, ik, jk);

Аргументы i_0, j_0, N, k ;

Результаты ik, jk ;

Примеч: $k[1:N]$ - последователь-
ность N кантований;

Нач Цел $i, i1, j1$;

$i1, j1 := i_0, j_0$;

Для $i := 1, \dots, N$ Вып

поворотик($i1, j1, k[i], i1, j1$);

$ik, jk := i1, j1$;

Кон;

Прим-----;
противопол($i1, i2$) $= ((i1 - i2) \bmod 3 = 3)$;

Примеч: логическая функция,
определяет, противоположны ли
грани $i1$ и $i2$;

Прим-----;
Проц опредповор4(i_0, j_0, ik, jk, k);

Примеч: определение последова-
тельности четырех кантований
 k , переводящей (i_0, j_0) в
 (ik, jk) (в случае, когда
четыре кантований достаточно);

Нач Цел $k1, k2, i1, j1$;

Для $k1 := 0, \dots, 3$ Вып

Для $k2 := 0, \dots, 3$ Вып

Если $(k1 - k2) \bmod 2 < 0$

Примеч: исключаем вырожденные
случаи;

То Нач $k[1], k[2] := k1, k2$;

поворотN($i_0, j_0, 4, k, i1, j1$);

Если $i1, j1 \neq ik, jk$

То Нач $k[3] := (k[1] + 2) \bmod 4$

$k[4] := (k[2] + 2) \bmod 4$

На выход

Примеч (из процедуры);

Кон;

Кон;

Вывод($0, 'т', 'Ошибка в обращении$
к "опредповор4":', '\ за четыре
кантования поворот невозможен!');

выход:

Кон;

Прим-----;

Проц вывповор(N, k);

Нач Цел i ;

Цел Масс $p[1:4, 1:10] =$

Текст(' Направо'),

Текст(' Вперед'),

Текст(' Налево'),

Текст(' Назад');

Примеч: массив из четырех строк
по 10 кодов символов (включая
кавычки) в каждой;

Для $i := 1, \dots, N$ Вып Вывод($0, 'т', p[k[$
 $i]$ ']']', '.')

Примеч: текстовый вывод в нулевой
канал;

Кон;

Прим-----;
Цел i_0, j_0, ik, jk ;

Цел Масс $k[1:4]$;

Прим-----;

Нач

Вывод('Введите (i_0, j_0) и (ik, jk) -
четверку чисел');

Ввод(i_0, j_0, ik, jk);

Если $i_0, j_0 \neq ik, jk$

То Вывод($0, 'т', 'Уже стоит как надо!'$)

Иначе

Если неразрешимо(i_0, j_0, ik, jk)

То Вывод($0, 'т', 'Для введенных$
данных задача неразрешима...')

Иначе

Нач

Вывод($0, 'т', 'Для решения$
задачи следует сделать:');

Если $i_0 = ik$ Или

противопол(i_0, ik)

То Нач $k[] := 0, 1, 2, 3$;

вывповор(4, k);

поворотN($i_0, j_0, 4, k, i_0, j_0$);

Примеч: выполняем

последовательность

четыре кантований,

приводящую на исходное место. Бывшая передней или задней грань i_0 становится примыкающей к передней;

Кон;
опредповор4(i_0, j_0, ik, jk, k);
вывповор(4, k);

Примеч: можно выполнить для контроля ответа поворотN($i_0, j_0, 4, k, ik,$

jk) и вывести (ik, jk);

Кон

Прим-----
Кон

В. ПРОХОРОВ

Задача предлагалась на практическом туре II Всесоюзной олимпиады по информатике (правда, с несколько иным условием).

В. КАСАТКИН, И. ПЕРЕХОД

Задача о сумме факториалов

Пусть необходимо данное натуральное N представить в виде суммы факториалов первых натуральных чисел.

Примером решенной задачи может быть представление числа 59 в виде суммы

$$59 = 2 \times 4! + 1 \times 3! + 2 \times 2! + 1 \times 1!$$

Поиски общего решения приводят к идее использовать факториальную систему счисления, для введения которой используем счеты нестандартного вида. Их анализ позволит установить все параметры факториальной системы счисления: определить базис системы, выяснить, как задаются верхние и нижние границы в каждом разряде, обдумать конструирование последовательного счетчика.

Работа на таких счетах ведется так же, как и на общепринятых. При последовательном счете косточка за косточкой сдвигаются влево и, когда все косточки на какой-нибудь проволоке окажутся сдвинутыми влево, осуществляется перенос единицы в следующий старший разряд.

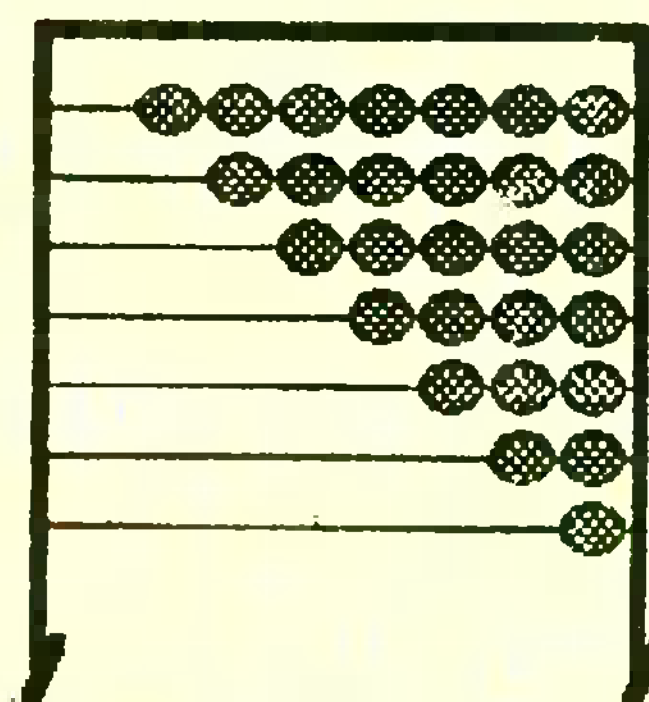
Особенность работы только в том, что количество косточек на разных проволоках разное. Обратим внимание, что в данной системе верхние границы каждого разряда фиксированы и неизменны. Величина верхней границы меняется от разряда к разряду: $\text{sup}(i) = i$. Нижние границы во всех разрядах одинаковы и равны нулю: $\text{inf}(i) = 0$.

Нетрудно заметить, что веса единиц разрядов образуют не геометрическую прогрессию, а последовательность факториалов первых натуральных чисел:

..., 720, 120, 24, 6, 2, 1,

или ..., 6!, 5!, 4!, 3!, 2!, 1!

Формула целого числа факториальной системы имеет вид:



$$a_n a_{n-1} \dots a_2 a_1 \phi = a_n \cdot n! + a_{n-1} \times (n-1)! + \dots + a_1,$$

где $\text{inf}(i) \leq a_i \leq \text{sup}(i)$.

Пример: $2121_\phi = 2 \times 4! + 1 \times 3! + 2 \times 2! + 1 \times 1!$

Из сказанного следует, что в факториальной системе существуют цифровая и «многочленная» форма записи чисел. Однако здесь многочлен не является суммой степеней основания системы, поэтому факториальная система называется не степенной, а смешанной.

К этой системе мы можем применить ранее рассмотренную методику конструирования последовательного счетчика, для этого достаточно использовать введенную в [1] процедуру обобщенного следования SUCC.

Ниже приводится вариант программы последовательного счета на Бейсике.

```

10 *   ФАКТОРИАЛЬНАЯ СИСТЕМА
20 *   N-РАЗРЯДНЫЙ СЧЕТЧИК
30 INPUT "N="; N : DIM A(N)
40 INF=0
50 FOR I=1 TO N
60 *   ВЫВОД
70 IF I=1 THEN
   FOR J=N TO 1 STEP -1:
     PRINT A(J); : NEXT J: PRINT
80 *   +1
90 SUP=I
100 IF A(I)=SUP THEN A(I)=INF
     ELSE A(I)=A(I)+1

```

```

110 IF A(I)<>INF THEN 50 ELSE NEXT I
120 END

```

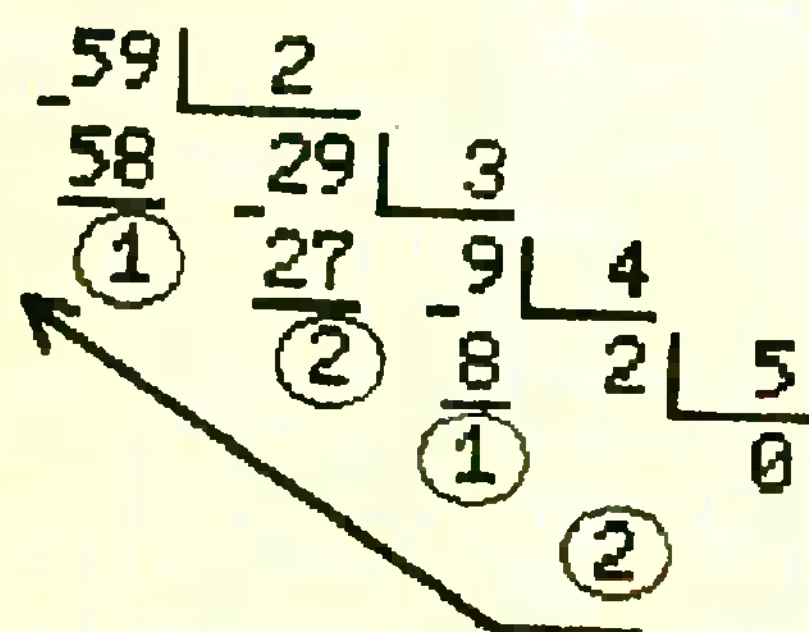
Задача, поставленная в начале статьи, решается в два этапа: сначала данное десятичное число N следует заменить равным ему числом факториальной системы, а затем, обратившись к полученному представлению числа N в факториальной системе, записать его в виде искомой суммы факториалов.

Первая часть задачи решается очень просто и красиво с использованием алгоритма последовательного деления в несколько измененной редакции. Чтобы данное целое число N десятичной системы заменить равным ему числом факториальной системы, достаточно последовательно делить его с остатком на 2, 3, 4, 5 и т. д., пока в частном не образуется ноль.

Полученные при таком последовательном делении остатки и являются цифрами искомого представления числа N в факториальной системе.

76

Пример: число 59 записать в факториальной системе.



$$59_{10} = \overline{2121}_0$$

Программа, реализующая алгоритм последовательного деления, имеет следующий вид.

```

10 ' ФАКТОРИАЛЬНАЯ СИСТЕМА
20 ' ДЕШИФРАТОР 10->!
30 INPUT "N=";N : K=N : D=2 : J=24
40 WHILE K<>0
50 ' ВЫВОД
60 C=K MOD D :
   LOCATE,J : PRINT C; : J=J-2
70 '-----
80 K=K\D : D=D+1
90 WEND : END

```

Второй этап не требует никаких вычислений — полученные остатки (цифры факториальной системы) образуют решение, они показывают, факториалы каких чисел следует включить в искомую сумму.

Программа решения задачи в целом приведена ниже.

```

10 ' ФАКТОРИАЛЬНАЯ СИСТЕМА
20 ' СУММА ФАКТОРИАЛОВ
30 INPUT "N=";N : K=N : D=2 : J=70
40 WHILE K<>0
50 ' ВЫВОД
60 C=K MOD D :
   IF C<>0 THEN LOCATE,J :
   PRINT "+"; C; "*"; D-1; "!"; : J=J-10
70 '-----
80 K=K\D : D=D+1
90 WEND : LOCATE,J+10 : PRINT " "
100 END

```

Литература

1. *Переход И., Касаткин В.* Комбинаторные задачи // Информатика и образование. 1990. № 6.
2. *Переход И., Касаткин В.* Обобщенный счетчик // Информатика и образование. 1991. № 2.

ВНИМАНИЮ ЧИТАТЕЛЕЙ!

Для учебных заведений (школ, техникумов, вузов)
научно-производственным объединением «Коминсервис»
подготовлен сводный каталог
программных средств (частично аннотированный)
с указанием стоимости и адресов тиражирующих организаций.

Каталог записан на импортном магнитном диске 5,25'.
Ориентировочная стоимость каталога на дискете 50—60 руб.
Заявки с указанием типа ЭВМ просим направлять по адресу:

Москва, п/о 121609, а/я 11.
Телефон для справок: 232-26-21.

Каталог одобрен секцией ССО НМС Гособразования СССР.

вами: 1-А, 2-Б, 3-В, 4-Г, 5-Д. Тогда задача примет следующий вид:

```

Б Г Д   А Б В
В А   →   Г Д

```

Теперь снова заменим буквы на цифры произвольным образом: А-5, Б-2, В-1, Г-3, Д-4.

Тогда эквивалентная 12-ходовая задача примет следующий вид:

```

2 3 4   5 2 1
1 5   →   3 4

```

По каждому классу эквивалентных задач, т. е. по задачам решаемым за 4, 6 и... ходов, можно составить 120 задач. Эквивалентные задачи позволяют рассматривать «Игру 5» как один из эффективнейших тестов, направленных на изучение процессов интеллектуального самообучения человека. Испытуемому (игроку) предъявляется серия эквивалентных задач (от 2 до 6) с разной степенью сложности. Учитывается время решения каждой задачи и количество ходов решения. Показатель интеллектуального самообучения оказывается тем выше, чем скорее игрок научается видеть эквивалентность задач и соответственно решать их быстрее и оптимальнее*.

Краткое описание программы

После инструкции на экране появляются два поля: «условие» и «цель». Дается подсказка, что перемещение фишек-цифр в поле «условие» осуществляется с помощью стрелок \rightarrow , \downarrow и т. д. Отказ от решения задачи в любом месте с помощью нажатия клавиши «ТАБ». Переход к решению следующей задачи осуществляется клавишей «пробел». Предлагаемый вариант про-

граммы позволяет решить только 5 задач: 4-, 6-, 8-, 10-, 12-ходовых. По желанию можно расширить количество предъявляемых задач, задав условия и цели в операторах DATA программы.

В связи с тем что важно лишь относительное время решения задач (по отношению друг к другу), оно подсчитывается условно с помощью простейшего счетчика, накапливающего единичку. После решения или просмотра всех пяти задач результативность выводится на экран дисплея, а также записывается на диск в файл с именем из четырех букв, набранных в латинском алфавите (двух первых букв имени и фамилии игрока). Результативность решения — число, соответствующее оптимальному количеству ходов для каждой задачи, число ходов, за которые она была реально решена, и время решения в условных секундах. В связи с тем что в программе происходит обращение к диску, в процессе работы с ней должна быть налажена локальная сеть.

```

10 CLEAR 1000
20 SCREEN 2
30 CLS
40 DIM A(5,6),O(5,6),Y(6),
      Z(6),Q(5)
42 DIM WK(5),M(5),MM(5),WF(5)
45 REM КООРДИНАТЫ ЦИФР
50 FOR X=1 TO 6
60 Y(X)=(X-INT((X-1)/3)*3)*5-1
70 Z(X)=(INT((X-1)/3)+3)*2
80 NEXT X
100 REM ОПРОС ИМЕНИ И ФАМИЛИИ
105 REM ДЛЯ СОЗДАНИЯ ВЫХОДНОГО
      ФАЙЛА
110 PRINT"(Пользуйтесь только
      шрифтом LAT)"
120 INPUT "Ваше имя : ";IM$
130 INPUT "Ваша фамилия : ";
      FAM$
140 FLN$=MID$(IM$,1,2)+
      MID$(FAM$,1,2)
160 CLS
170 FOR S=1 TO 5
180 FOR I=1 TO 6
190 READ A(S,I)
200 IF A(S,I)=0 THEN M(S)=I
210 NEXT I
220 READ Q(S)
230 FOR I=1 TO 6
240 READ O(S,I)
250 IF O(S,I)=0 THEN MM(S)=I

```

```

260 NEXT I
270 NEXT S
280 FOR S=1 TO 5
290 GOSUB 1520
300 COLOR 2,2,7
310 PRINT " УСЛОВИЕ : ";
      TAB(19);"ЦЕЛЬ : "
320 COLOR 8,2,2
330 PRINT AT(18,14) "ЗАДАЧА : "
      ;S
340 PRINT AT(24,11)
      " ┌───┬───┬───┬───┐"
350 PRINT " ПОЛЬЗУЙТЕСЬ
      КЛАВИШАМИ | | | | | | |"
360 PRINT AT(24,13)
      " ┌───┬───┬───┬───┐"
370 PRINT AT(44,14) "1234"
380 PRINT "ЕСЛИ ВЫ НЕ МОЖЕТЕ
      РЕШИТЬ ЗАДАЧУ, НАЖМИТЕ [TAB]"
390 PRINT AT(44,16) " ┌───┐"
400 RESTORE 1090
410 FOR X=1 TO 12
420 READ LA, LB, LC, LD
430 LINE (LA, LB)-(LC, LD)
440 NEXT X
520 GOSUB 1390
540 T$=INKEY$
550 WW=WW+1
560 IF T$="" THEN GOTO 540
570 IF ASC(T$)=9 THEN GOTO 790
580 IF ASC(T$)<>27 THEN
      GOTO 460
590 T$=INKEY$
600 IF T$="" THEN GOTO 590
610 ON ASC(T$)-64 GOTO
      1050,1110,1170,1230
620 WR=WR+1
670 FOR H=1 TO 6
680 IF A(S,H)<>O(S,H) THEN
      GOTO 450
690 NEXT H
700 WF(S)=(WW/100+.5)/10
710 WW=0
720 WK(S)=WR
730 WR=0
740 FOR D=1 TO 12
750 PRINT AT(O,D) SPC(60)
760 NEXT D
770 BEEP
780 IF S=5 THEN GOTO 1640
790 PRINT AT(0,20) "#####
      #####
      #####"
800 PRINT "Если вы готовы к
      следующей задаче, нажмите
      <ПРОБЕЛ>"
810 B$=INKEY$
815 IF B$="" THEN GOTO 810
818 CLS
820 NEXT S
850 REM ДАННЫЕ ДЛЯ ЗАДАЧ
860 DATA 4,1,3,0,2,5,4
870 DATA 1,2,3,0,4,5
880 DATA 5,2,4,0,1,3,6
890 DATA 2,4,3,0,5,1
900 DATA 1,2,4,0,5,3,8
910 DATA 1,2,3,0,4,5
920 DATA 2,1,4,0,3,5,10
930 DATA 5,3,1,0,2,4
940 DATA 1,4,3,0,2,5,12
950 DATA 5,2,1,0,3,4
960 REM КООРДИНАТЫ ДЛЯ СТРЕЛОК
970 DATA 271,120,271,127,311,
      121,311,128

```

* Подробнее см. Д. А. Поспелов, В. Н. Пушкин Мышление и автоматы. М.: Радио. 1972

```

980 DATA 343,124,358,124,384,
      124,399,124
990 DATA 268,125,271,128,271,
      128,274,125
1000 DATA 308,123,311,120,311,
      120,314,123

520 GOSUB 1390
540 T$=INKEY$
1010 DATA 356,127,359,124,359,
      124,356,121
1020 DATA 386,127,383,124,383,
      124,386,121
1050 IF M(S)>3 THEN GOTO 460
1060 A(S,M(S))=A(S,M(S)+3)
1070 M(S)=M(S)+3
1080 A(S,M(S))=0
1090 GOSUB 1390
1100 GOTO 540
1110 IF M(S)<4 THEN GOTO 460
1120 A(S,M(S))=A(S,M(S)-3)
1130 M(S)=M(S)-3
1140 A(S,M(S))=0
1150 GOSUB 1390
1160 GOTO 540
1170 IF M(S)=3 OR M(S)=6 THEN
      GOTO 460
1180 A(S,M(S))=A(S,M(S)+1)
1190 M(S)=M(S)+1
1200 A(S,M(S))=0
1210 GOSUB 1390
1220 GOTO 540
1230 IF M(S)=3 OR M(S)=6 THEN
      GOTO 460
1240 A(S,M(S))=A(S,M(S)+1)
1250 M(S)=M(S)+1
1260 A(S,M(S))=0
1270 GOSUB 1390
1280 GOTO 540
1330 LOCATE 0,5
1340 PRINT "  |  |  |  |  |  "
1350 PRINT " |  |  |  |  |  "
1360 PRINT " |  |  |  |  |  "
1370 PRINT " |  |  |  |  |  "
1380 PRINT "  |  |  |  |  |  "
1390 FOR X=1 TO 6
1400 IF X=M(S) THEN GOTO 1490
1410 PRINT AT(Y(X),Z(S)) A(S,X)
1420 NEXT X
1430 RETURN
1460 RETURN
1470 LOCATE 0,5
1480 PRINT TAB(28) "  |  |  |  |  |  "
1490 PRINT TAB(28) " |  |  |  |  |  "
1500 PRINT TAB(28) " |  |  |  |  |  "
1510 PRINT TAB(28) " |  |  |  |  |  "
1520 PRINT TAB(28) "  |  |  |  |  |  "
1530 FOR X=1 TO 6
1540 IF O(S,X)=0 THEN GOTO 1620
1550 PRINT AT(Y(X)+25,Z(S))
      O(S,X)
1560 NEXT X
1570 RETURN
1580 FOR X=1 TO 6
1590 IF O(S,X)=0 THEN GOTO 1620
1600 PRINT AT(Y(X)+25,Z(S))
      O(S,X)
1610 NEXT X
1620 RETURN
1630 CLS
1650 COLOR 2,2,7
1660 PRINT AT(25,2) "ТЕСТ
      ОКОНЧЕН"
1670 COLOR 8,2,2
1680 OPEN FLN$ FOR OUTPUT
1690 PRINT# IM$; " ";FAM$
1700 FOR O=1 TO 5
1710 PRINT#(O+1)*2;" -ходовая
      задача:"
1720 IF WK(O)=0 THEN GOTO 1750
1730 PRINT# WK(O);" ходов ";
      WF(O);" секунд"
1740 GOTO 1760
1750 PRINT# " < НЕ РЕШЕНА >"
1760 NEXT O
1770 CLOSE
1780 *FOR O=1 TO 5
1790 PRINT (O+1)*2;" -ходовая
      задача:"
1800 IF WK(O)=0 THEN GOTO 1750
1810 PRINT WK(O);" ходов ";
      WF(O);" секунд"
1820 GOTO 1760
1830 PRINT " < НЕ РЕШЕНА >"
1840 NEXT O
1850 END

```

Структура программы

Строки 22-28 — инструкция (приведена выше).
 Строки 50-80 — подсчет координат X, Y для расположения цифр-фишек в полях «условие» и «цель».
 Строки 100-140 — организация имени выходного файла.
 Строки 160-270 — считывание конкретных значений расположения фишек-цифр в полях «условие» и «цель», а также числа оптимальных

ходов решения каждой задачи. По увеличении количества задач в программе следует изменить соответствующее значение переменной S (строка 170 и 280).
 Строки 280-520; 1330-1460; 1470-1580 — оформление экрана.

Строки 540-650 — счетчик ходов и времени решения.
 Строки 670-820 — оформление перехода к следующей задаче.

Строки 850-950 — конкретные данные для условий и целей каждой задачи, а также величина оптимального числа ходов каждой задачи.

Первая DATA — шесть цифр — условие, седьмая цифра — оптимальность ходов решения, вторая DATA — шесть цифр — цель.

Строки 970-1030 — координаты начала и конца отрезков изображения стрелок для указания направлений перемещения фишек-цифр. Используются только в операторе LINE в строке 510. (В случае отсутствия «совмещенного» Бейсика этот оператор можно безболезненно сократить.)
 Строки 1050-1300 — организация перемещения цифр-фишек в поле «условие».

Строки 1630-1840 — вывод результатов в выходной файл и на экран.

Программа написана учеником X класса школы № 8 Москвы В. Харитонюком под руководством канд. психол. наук О. Ю. Ермолаева.

**В. ХАРИТОНЮК,
О. ЕРМОЛАЕВ**

79

Сообщаем нашим читателям,

что разработка В. Завалишина «J-LISP», о которой мы сообщали в «ИНФО» № 2 за 1991 г., была представлена на выставке «Новые педагогические программные средства», проходившей на ВДНХ СССР в 1991 г., и завоевала Золотую медаль ВДНХ СССР и Диплом первой степени.

Координаты курсора хранятся в ячейке 156 (здесь и далее используются восьмеричные числа, за исключением отмеченных случаев). Значение ее младших 6 битов (0—5) определяют координату по горизонтальной оси (от 0 до 77), значения битов 6—10₁₀ — по вертикальной. В режиме 32₁₀ символа в строке позиция курсора на горизонтальной оси (0—31₁₀) определяется целой частью от деления на 2 точной его координаты.

Ячейка 160 определяет адрес того байта экранной памяти, с которого начнется вывод очередного символа. Он соответствует текущему положению курсора.

В режиме ГРАФ (в байте 50 записан не ноль) курсор указывает на отдельные адресуемые точки. В этом случае содержимое ячейки 156 показывает, в какой области из $8 \times 10_{10}$ (в режиме 64₁₀ символа в строке) или $4 \times 10_{10}$ (в режиме 32₁₀ символа в строке, когда точку определяют два соседних бита экранной памяти) точек находится курсор. Координаты, записанные в этой ячейке, указывают на место, в котором курсор будет находиться при выходе из графического режима. Положение графического курсора внутри такой области определяют ячейки 154, 166, 170 и 174.

Младший байт ячейки 154 определяет положение точки, в которой находится курсор, на горизонтальной оси. Если в бите 9 этой ячейки записана единица, то текущий режим — 32₁₀ символа в строке (256₁₀ точек по горизонтали), если 0 — то 64₁₀ символа (512₁₀ точек).

В последнем режиме номер установленного (т. е. того, в котором записана единица) бита младшего байта ячейки 154 указывает

координату курсора внутри «области символьного курсора» (ОСК) по горизонтали. Так, если содержимое ячейки 154 равно ...010001000, то графический курсор указывает на пятую по горизонтали точку внутри той ОСК, в которой он находится. В режиме 1 координата определяется установкой пары битов (0—1, 2—3, 4—6 или 5—7). Если, например, в ячейке 154 записано ...1111000000, то курсор находится в четвертой по горизонтали точке ОСК.

Младший байт ячейки 154 логически умножается на маску цвета, а затем логически складывается с тем байтом экранной памяти, «где находится» графический курсор; адрес этого байта определяется содержимым ячейки 170. В 6 младших битах ячейки 166 записана координата ОСК по горизонтальной оси (та же, что в 6 младших битах ячейки 156).

Теперь о вертикальной координате. Биты 6—13₁₀ ячейки 166 определяют координату Y графического курсора (от 0 до 357, т. е. до 239₁₀). Биты 6—9 содержимого ячейки 174 определяют вертикальную координату точки внутри ОСК. Кроме того, адрес, указанный в ячейке 170, может однозначно определить вертикальную координату графического курсора, если учесть рулонный сдвиг.

При переходе в графический режим содержимое ячейки 106 изменяется с 20 000 на 2000, а ячейки 224 —

с 0 на 2. Назначение этих изменений неясно.

Кроме того, мне удалось установить, где хранится информация о табулируемых позициях (устанавливаемых клавишами **СУ+М** и снимаемых клавишами **СУ+Р**). Ячейки 112—120 определяют их следующим образом:

ячейка 112 хранит информацию о позициях 0—15₁₀;

114 — 16₁₀—31₁₀;

116 — 32₁₀—47₁₀;

118 — 48₁₀—63₁₀.

(последние две ячейки задействованы только в режиме 64₁₀ символа в строке). Установленные биты задают позиции табуляции (например, бит 2 ячейки 112 — позицию 2, бит 2 ячейки 114 — позицию 18 и т. д.).

Ячейка 214 определяет цвета символа, 212 — цвет фона, 220 — цвет символа в служебной строке, 216 — цвет фона в служебной строке. О том, как задается цвет в БК, уже говорилось на страницах «ИНФО». Подчеркну, что изменение значений в двух последних ячейках не изменяет цвет служебной строки, а лишь указывает режим вывода в ней подсказок РУС, ЛАТ, ПОДЧ, ИНВ, ИСУ, БЛР.

В ячейке 176 записана координата X последней выведенной (или стертой) по запросам ЕМТ 30 или ЕМТ 32 графической точки, а в ячейке 200 — ее координата Y. Они используются при построении отрезков по запросу ЕМТ 32 для указания

И. ХРИСТОФОРОВ

Ниже следует внутренняя рецензия на статью И. Христофорова. Обычно рецензия остается известной только редактору, однако в данном случае она весьма информативна. Кроме того, представление о редакционной «кухне» может быть полезным молодым авторам, поэтому мы и впредь будем публиковать интересные рецензии.

Автором проделана значительная работа по экспериментальному определению назначения ячеек системной области БК-0010, результаты которой в основном изложены правильно. Кое в чем автор неточен, или, вернее, неправильно формулирует свои выводы. Например, ячейка 154 рассматривается им как слово, хотя из текста драйвера дисплея БК-0010 очевидно, что это два отдельных байта, 154 и 155. Если еще назвать их «по имени», т. е. сказать, что байт 154 — маска позиции графического курсора (в пределах байта), а байт 155 — начальная маска графического курсора (или, иначе выражаясь, графической точки), то положение значительно прояснится. При описании ячейки 166 создается впечатление, что ее содержимое чуть ли не дублирует содержимое ячейки 156; между тем ячейка 166 задает номер (адрес) графической точки на экране, а 156 — номер символа. Ячейка 156 работает и в символьном, и в графическом режиме, определяя номер «рабочего» зна-

коместа, а ячейка 166 — только в графическом режиме (при возврате в символьный режим ее содержимое фиксируется).

Автор затрудняется объяснить назначение ячеек 106 и 224 и причины изменений их содержимого при смене режимов. Для ясности можно сообщить, что 106 — буфер константы повтора (время цикла задержки в режиме «повтор»), а 224 — счетчик индикаторов служебной строки, или, иначе говоря, условный номер выводимого в служебную строку в графическом режиме текста. Содержимому ячейки 224, равному 0, соответствуют пустые позиции 20_{III}—23_{III} (считая с правого края строки), 2 — индикатор ГРАФ, 3 — ЗАП, 4 — СТИР.

Ячейки 112—120 определяются в тексте драйвера как блок из 8 байтов, носящий название «маска табуляции».

Несмотря на указанные недостатки, материал представляет заметный интерес для читателей и может быть опубликован.

Ю. ЗАЛЬЦМАН

Программа «Экспертные оценки» для школьной психологической службы

Программа «Экспертные оценки» написана на стандартном Бейсике для ПЭВМ УКНЦ. Она легко модифицируется, с ее помощью можно проводить различные варианты экспертных оценок.

Собственно термин «экспертная оценка» в условиях современной школы практически не употребляется, да и работа по организации экспертной оценки (в ее различных вариантах)

в школах проводится чрезвычайно редко в силу разных причин. Одна из них — сложная организационная сторона проведения экспертиз. Требуется, в частности, большое время на изготовление экспертных бланков, при значительном количестве материала затруднен его анализ и т. п. Предлагаемая ниже программа позволяет быстро решить разнообразные технические и организацион-

ные вопросы, оперативно осуществить экспертную оценку.

Следует сказать несколько слов о ее содержании. По-видимому, в ближайшее время подобного рода процедуры станут неотъемлемой частью в структуре целостного школьного учебного процесса. Экспертная оценка будет играть роль одного из механизмов обратной связи в живом школьном организме. Кто же будет экспертом и что он будет оценивать? На первый вопрос следует однозначно ответить, что экспертами будут и ученики, и учителя. А вот содержание экспертных оценок, по-видимому, безгранично. Это и уровень преподавания, и эффективность программы, и удобство расписания, и организация свободного времени учащихся школы, и многое другое. В предлагаемой ниже программе предлагается оценить 10 качеств личности учащегося. Это такие личностные качества, как трудолюбие, доброта и т. п. Экспертами могут выступать и ученики и учителя, и оценивать можно и тех и других. Конечно же, можно производить с помощью настоящей программы экспертную оценку и по иным параметрам, что потребует изменений в соответствующем операторе DATA программы, а также в подписях к результирующим графикам.

Описание программы.

Программа работает в SCREEN1, SCREEN2 — не используется, в связи с чем все рисунки построены с помощью псевдографических символов.

В программе предусмотрены экспертная оценка любого числа учащихся, а также произвольное количество экспертов. Номер эксперта и фамилия оцениваемого постоянно указаны на экране.

Экспертная оценка производится по принципу шкалирования. На экране дисплея появляется вертикальная шкала с горизонтальными делениями, пронумерованными от 0 до 10. Параллельно шкале «бегают» стрелка-указатель. Эксперт останавливает ее напротив того числа шкалы, которое в качестве экспертной оценки он ставит данному ученику. Например, эксперт № 1 оценивает ученика Иванова по качеству «трудолюбие» и ставит ему оценку «5». Для этого он подводит стрелку-указатель к цифре 5 на шкале и нажимает клавишу «ВВОД». Перемещение стрелки-указателя осуществляется клавишами

82

↑ и ↓.

После того как первый эксперт оценил все 10 качеств личности ученика № 1, он оценивает ученика № 2 и так до конца. Затем наступает очередь второго эксперта, который также оценивает всех учащихся по всем качествам, и т. д.

После окончания процедуры оценивания на экран дисплея выдаются обобщенные графики каждого учащегося. Это так называемый профиль ученика у данного эксперта по всем 10 качествам. Иными словами, это 10 вертикальных шкал с горизонтальными делениями, отмеченными цифрами от 0 до 10. На них же наносятся звездочки «*», указывающие весь «маршрут».

```
1 COLOR 8,1,1
10 JJ=1
20 J=1
22 X=1
30 Z=1
40 F=1
50 CLS
60 INPUT "Введите кол-во фамилий";A
70 INPUT "Введите кол-во экспертов";X
80 DIM B$(A),D$(10),E(10,A,X)
90 CLS
100 FOR C=1 TO A
110 ? "ФАМИЛИЯ N ";C;
120 INPUT B$(C)
130 NEXT
```

```
140 DATA ТРУДОЛЮБИВЫЙ,
    ОТВЕТСТВЕННЫЙ, ДОБРЫЙ,
    ДИСЦИПЛИНИРОВАННЫЙ,
    КОЛЛЕКТИВИСТ, НАСТОЙЧИВЫЙ,
    СПОКОЙНЫЙ, ТВОРЧЕСКИЙ,
    ОТЗЫВЧИВЫЙ, УРВНОВЕШЕННЫЙ
```

```
170 FOR C=1 TO 10
180 READ CC$
190 D$(C)=CC$
200 NEXT
210 CLS
215 ? AT(5,2);" ОЦЕНИВАЕТ
    ЭКСПЕРТ ";Z
220 ? AT(5,5);" —"
230 FOR C=6 TO 15
240 ? AT(5,C);" —"
250 NEXT
260 ? AT(5,15);" —"
270 FOR C=5 TO 15
280 ? AT(1,C);CCC
290 CCC=CCC+1
300 NEXT
310 Y=5
320 ? AT(25,5);"КАЧЕСТВО:";
    D$(JJ);" "
330 ? AT(25,7);"ОЦЕНИВАЕМ ";
    B$(J);" "
340 A$=INKEY$
350 ? AT(9,Y);CHR$(130)
360 ? AT(9,Y-1);" "
370 ? AT(9,Y+1);" "
380 IF A$=CHR$(65) THEN Y=Y-1
390 IF Y<5 THEN Y=5
400 N=Y
410 IF A$=CHR$(66) THEN Y=Y+1
420 IF Y>15 THEN Y=15
430 IF A$=CHR$(13) THEN 450
440 GOTO 340
450 E(JJ,J,Z)=N

460 JJ=JJ+1
470 IF JJ=11 THEN 490
480 GOTO 320
490 JJ=1
500 J=J+1
510 IF J>A THEN 521
520 GOTO 320
521 Z=Z+1
522 JJ=1
523 J=1
524 ? AT(9,Y);" "
525 CCC=0
529 IF Z>X THEN 530 ELSE 215
530 CLS
531 Z=1
535 ? AT(5,5);
    "Т Т Т Т Т Т Т Т Т Т"
560 FOR EEE=6 TO 15
570 ? AT(5,EEE);
    " + + + + + + + + + + "
580 NEXT
590 ? AT(5,15);
    " — — — — — — — — — — "
540 COLOR 6,1,1
600 FOR EEE=5 TO 15
610 ? AT(0,EEE);SSS
620 ? AT(43,EEE);SSS
630 SSS=SSS+1
640 NEXT
650 ? AT(5,16);"А В С D E F
    G H I J"
670 ?AT(50,5);""А'-ТРУДОЛ."
680 ?AT(50,6);""В'-ОТВЕТ."
690 ?AT(50,7);""С'-ДОБРЫЙ."
```

```
700 ?AT(50,8);""D'-ДИСЦИП."
710 ?AT(50,9);""E'-КОЛЛЕКТ."
720 ?AT(50,10);""F'-НАСТОЙ."
730 ?AT(50,11);""G'-СПОКОЙ."
740 ?AT(50,12);""H'-ТВОРЧ."
750 ?AT(50,13);""I'-ОТЗЫВЧ."
760 ?AT(50,14);""J'-УРАВН."
770 ?AT(0,3);"—————"
780 R=6
790 ? AT(10,2);" ГРАФИК ";
    B$(F);" ОЦЕН. ЭКСПЕРТ N ";Z
800 COLOR 8,1,1
810 FOR TTT=1 TO 9
820 ?AT(R+3,E(TTT,F,Z));"X"
830 ?AT(R-1,E(TTT,F,Z));"X"
840 ?AT(R,E(TTT,F,Z));"X"
850 ?AT(R+1,E(TTT,F,Z));"X"
860 ?AT(R+2,E(TTT,F,Z));"X"
870 FOR LLL=E(TTT,F,Z) TO
    E(TTT+1,F,Z)
880 ? AT(R+3,LLL);"X"
890 NEXT
900 FOR LLL=E(TTT,F,Z) TO
    E(TTT-1,F,Z)
910 ? AT(R-1,LLL);"X"
920 NEXT
930 R=R+4
940 NEXT
950 IF INKEY$="" THEN 950
960 F=F+1
961 SSS=0
970 IF F>A THEN 990
980 GOTO 540
990 COLOR 8,1,1
1000 Z=Z+1
1010 R=6
1020 F=1
1030 SSS=0
1040 IF Z>X THEN CLS
    ELSE 540
1050 LOCATE 0,0
```

Структура программы. Строки 215-300 — построение оценочной шкалы на экране.

Строки 340-440 — управление стрелками.

Строки 450-529 — постановка баллов.

Строки 531-790 — построение результирующих графиков.

Строки 800-940 — построение индивидуального «профиля», «маршрута» экспертной оценки каждого ученика.

Строки 950-1040 — смена графиков и профилей.

Программа выполнена учеником XI класса школы № 469 Москвы Д. Полянским.

Д. ПОЛЯНСКИЙ,
О. ЕРМОЛАЕВ

Учебно-игровая программа «Декартовы координаты»

Идея написания программы по теме «Декартовы координаты» возникла в прошлом учебном году. В образовавшемся творческом коллективе постановщиком задачи была учительница математики, консультантом — преподаватель информатики и программистом — автор.

Программа-тренажер реализована на Бейсике для ПЭВМ «Корвет» (после небольшого редактирования она может работать на ПК «Ямаха»).

Цель программы — помочь учащимся VI класса в изучении математической темы «Декартовы координаты», сделав ее более наглядной и интересной.

Программа очень проста в управлении, поэтому инструкция, с которой начинается работа пользователя, уместается в одном кадре.

В следующем кадре появляется меню. Программа состоит из двух частей:

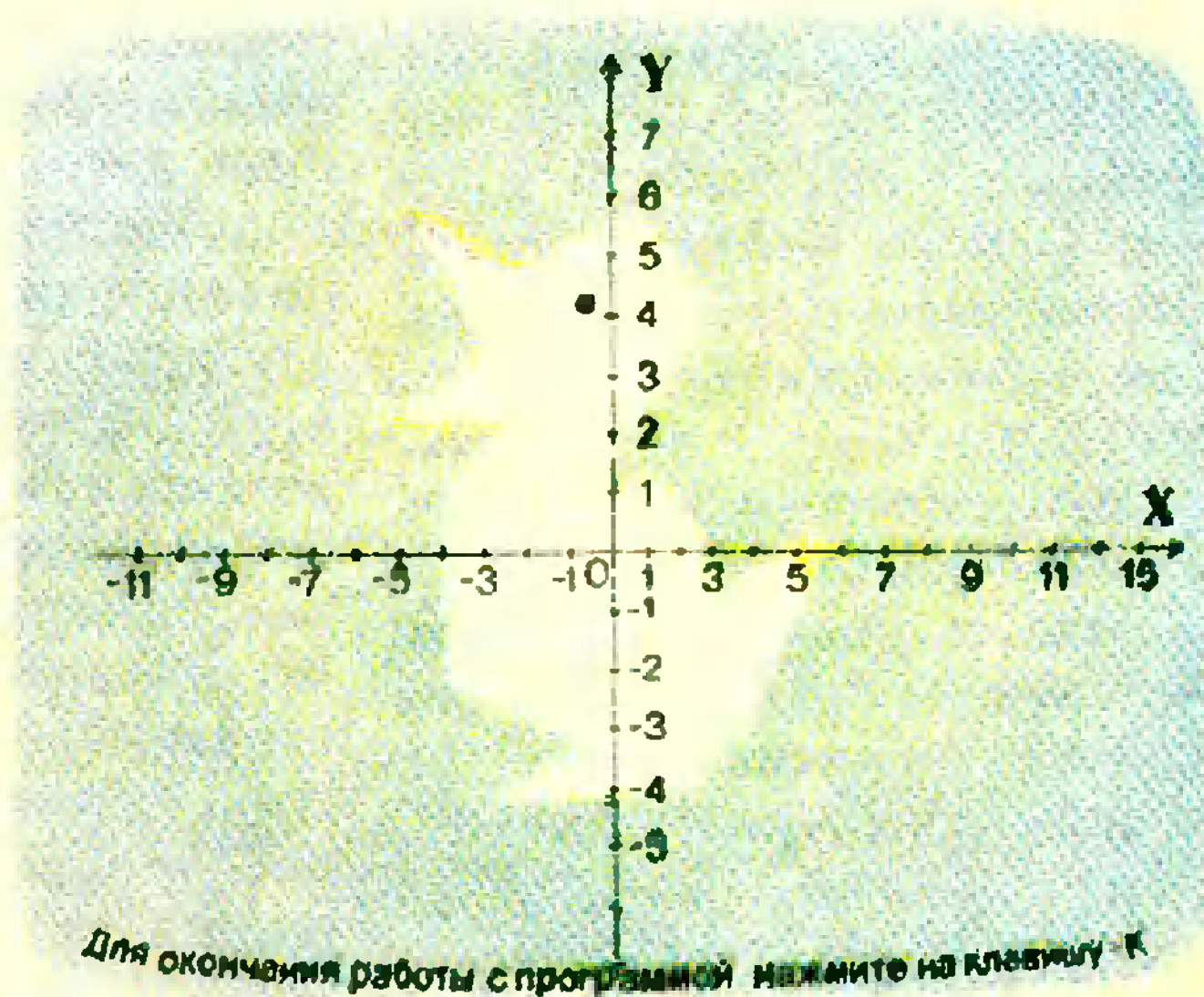
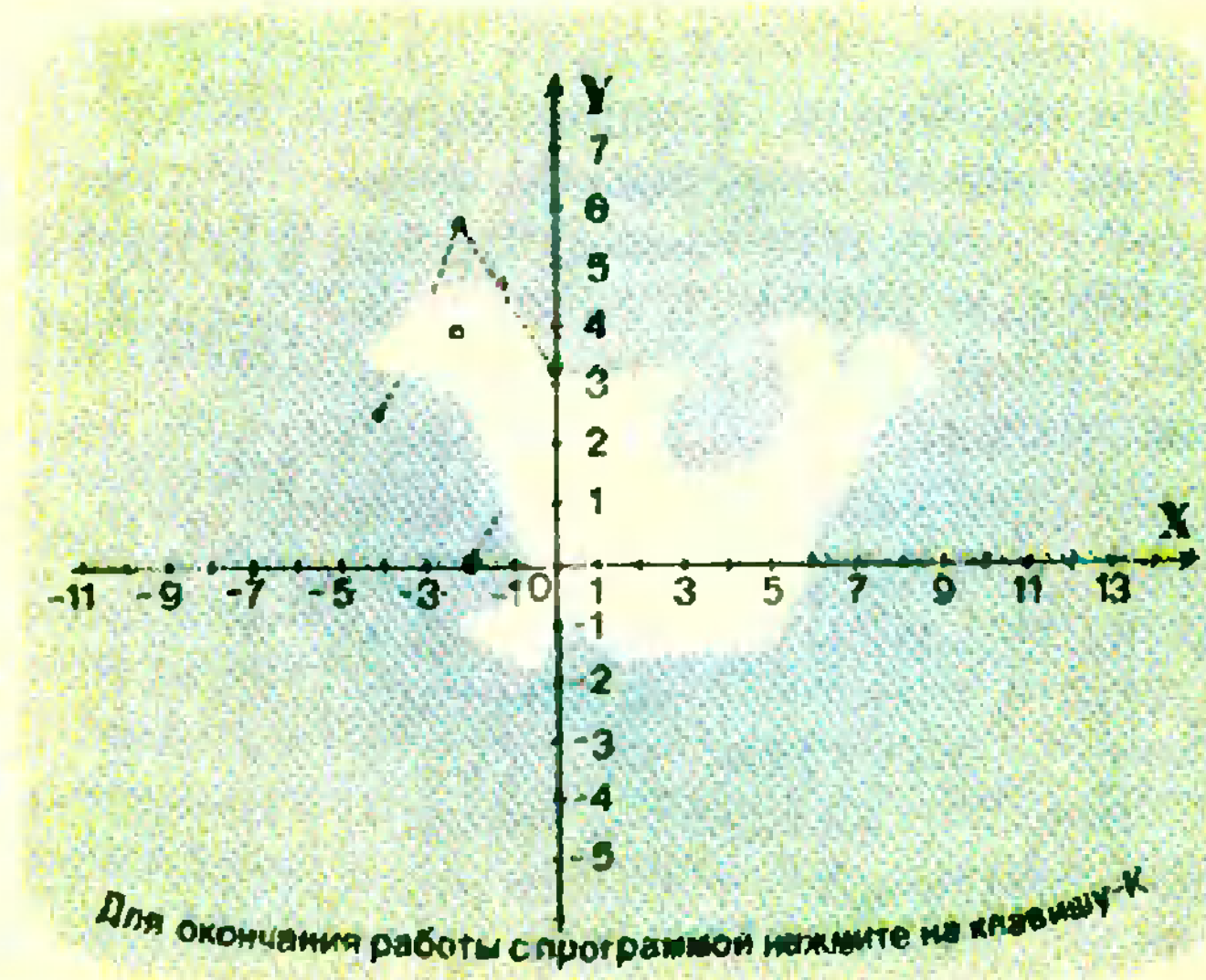
1. Определение координат точек, возникающих на экране.

2. Получение изображений животных по заданным координатам.

Первая часть программы реализована в виде игры. Точки — это метеориты, падающие на Землю и уничтожающие ее. Для спасения Земли необходимо правильно указать координаты метеоритов. Для выполнения задания даются три попытки. Если координаты точки указаны верно, то она исчезает (в противном случае остается) и на экране появляется следующая. Всего точек пять. По конечному результату (количеству оставшихся на экране точек) учитель может определить уровень усвоения данной темы каждым учеником.

Цель работы второй части программы — получить правильное изображение картинки по заданным координатам, выбрав одну из четырех: лебедя, тюленя, белку или утенка.

На экране появляется система координат, в которой задана точка в верхнем правом углу. Управляя курсором (с помощью стрелок: вправо, влево, вверх, вниз), ученик ставит свою точку. После этого появляются ее координаты, курсор перемещается в начало и т. д. Все точки соединяются линиями. Итог работы — изображение выбранной картинки, закрашенное цветом. Если линии, прочерченные учеником и компьютером, совпадут — задание выполнено верно, в противном случае наглядно



видны ошибки. Открытость программы позволяет расширять ее, включая новые изображения животных.

Программа успешно используется в школе № 63 г. Самары.

Л. ВОЛЬПИН,
ученик XI класса

ЧТО МОЖЕТ ЭВМ?

RISC Laptop

Для постоянных читателей рубрики «Что может ЭВМ?», думаю, не представляют загадки два слова, вынесенные в заголовок. Ведь до недавнего времени архитектура RISC, основной особенностью которой является использование очень малого числа предельно коротких команд, вызвала бурные дебаты: быть или не быть?

Тем временем Toshiba выпустила на рынок переносной компьютер с автономным источником питания, работать с которым можно, положив его на колени. Новый лэптоп помимо необычной архитектуры имеет высокоскоростной электролюминесцентный дисплей. На экране с диагональю 11,5 дюйма можно сменять изображение, состоящее из 1152 × 900 пиксел, каждые 2 миллисекунды. А иначе за новым процессором с производительностью 13,2 млн. операций в секунду и не поспеть.

Соответствуют характеристикам процессора и оперативная память — 8М байт с возможностью расширения до 40М байт, и жесткий диск — 180М байт, и возможность наращивания внешних дисков до суммарной емкости 2,8Г байт.

О совместимости с IBM PC фирма умалчивает. Сообщается, правда, что в инструментальной системе нового RISC лэптопа предусмотрена возможность работы с файлами формата IBM PC.

Назад, к перу

Далеко не все пользователи достаточно ловко обращаются с клавиатурой. Учитывая это, фирма Apple (компьютеры «Apple» и «Macintosh») сделала ставку на так называемый графический интерфейс, в котором манипулировать приходится в основном не клавишами, а мышью.

Но даже в «Макинтош» приходится вводить текст. А интерфейс MS-DOS и вовсе не графический. Так что проблема остается.

Очередная попытка решить эту задачу предпринята компанией Grid Systems. Ее компьютер Grid Pad клавиатуры не имеет. Совсем. Для ввода текста используется электронное перо, которым пишут на специальном планшете. К сожалению, почерки Grid Pad разбирает пока плохо, предпочитает печатные буквы, да и их путает в 5 % случаев. Впрочем, по мере освоения пользователем специфики работы с этой моделью вероятность правильного распознавания повышается.

С одной стороны, не так плохо для «малыша» (вес компьютера вместе с планшетом — 2 кг), а с другой... Не проще ли потратить время на освоение клавиатуры, чем на улучшение почерка?

Понравилось — плати!

Именно это подразумевает термин shareware. Он обозначает программы, распространяемые на первый взгляд бесплатно, например через общедоступные «доски объявлений» компьютерных сетей или специальные фирмы, где оплачивают только дискету и работу по записи на нее. Предполагается, однако, что, «попробовав» программу и решив использовать в своей работе, честный пользователь купит ее, выслав автору объявленную сумму.

Честны не все (даже в Америке), и авторы shareware используют для стимулирования достойного поведения разнообразные ухищрения. Это, например, «экранная совесть» (заставка в начале программы, напоминающая о необходимости оплаты), распространение неполноценных версий (хочешь вполне работоспособную версию — плати автору вперед; впрочем, такой метод считается неэтичным), обеспечение заплатившего пользователя печатным руководством и консультациями.

Не подумайте, что так распространяются лишь любительские самоделки. Среди программ shareware много весьма авторитетных (для примера назовем хотя бы Microsoft Windows), а авторы объединены в Ассоциацию профессионалов.

Вот бы и нашим программистам (особенно работающим на домашних компьютерах типа БК-0010) такое! Почти бесплатное распространение (рубль за программу — не деньги) многими кооперативами уже налажено. Осталось «наладить» ответственность пользователей и отработать систему оплаты авторам.

Ключ или замок?

Программные средства защиты от копирования принципиально не могут обеспечить вполне эффективную защиту от «компьютерных пиратов». При достаточной квалификации любой блок защиты на дискете можно найти и нейтрализовать. Поэтому все чаще для охраны программ используются аппаратные средства.

Известная в этой области фирма Software Sekurity Inc. предлагает небольшие блоки, подключаемые к параллельному порту ПК. Работоспособность порт не теряет — к этому блоку можно, в свою очередь, подключать периферийные устройства. Надежность охраны как для разработчиков программ, так и для пользователей гарантируется.

Но интересно: почему фирма называет этот блок ключом?

Если не нравится изображение

Обработка изображений практически перестала быть компьютерной прерогативой и превратилась в самостоятельную индустрию. Одна из японских фирм предлагает очень дешевый (20 долларов), но вполне приличный графический процессор, который может быть встроен не только в компьютер, но и в обычный телевизор.

С помощью такого процессора вы можете уменьшать или увеличивать изображение на экране телевизора, поворачивать и инвертировать его.

Видеопроцессор может найти применение и в тех случаях, когда неисправен сам телевизор. С помощью простого программирования можно добавить недостающий цвет, уменьшить бочкообразность и даже масштабировать и приостанавливать наиболее привлекательные для вас места телепередачи.

Я. МАРГОЛИС, А. ИВАНОВ

Шестилетки: к творчеству через компьютер

...Мечтая, будь осторожен!..
Мечты иногда сбываются...

Морин Хэрли

Развитие детей дошкольного возраста с помощью работы на компьютерах, как свидетельствует отечественный [1] и зарубежный опыт [2, 3], является одним из важных направлений современной педагогики. В этой связи актуальными становятся вопросы о формах и методах обучения детей шестилетнего возраста. Настоящая статья посвящена этой проблеме и отражает опыт обучения шестилетних детей детского сада № 132 на базе школы № 63 г. Самары.

Наша педагогическая концепция обучения направлена на развитие различных творческих способностей детей дошкольного возраста. Сложность поставленной задачи определяется тем, что, с одной стороны, необходимо стремиться к развитию творческих способностей детей, а с другой, давать им знания о мире современных компьютеров в увлекательной, интересной форме. Основная цель этих занятий — научить растущего человека самостоятельно мыслить, развить фантазию и практически воплощать свои творческие идеи с помощью компьютера.

«Многие рассказы, повести и даже романы для взрослых начинаются со слов: «В уездном городе N...», а многие сказки для детей начинаются словами: «Жили-были...». Так как наша сказка «Компьютерная школа»*, фрагменты которой будут приведены в статье, написана и для детей, и для родителей, то начнем ее словами: «В уездном городе N жили-были...». Мы вам расскажем о самых интересных жителях города N.

А самыми интересными жителями в уездном городе N были два друга: Любознашкин и Хвастунишкин. Как вы уже догадались из их имен, Любознашкин был очень любознательным и инте-

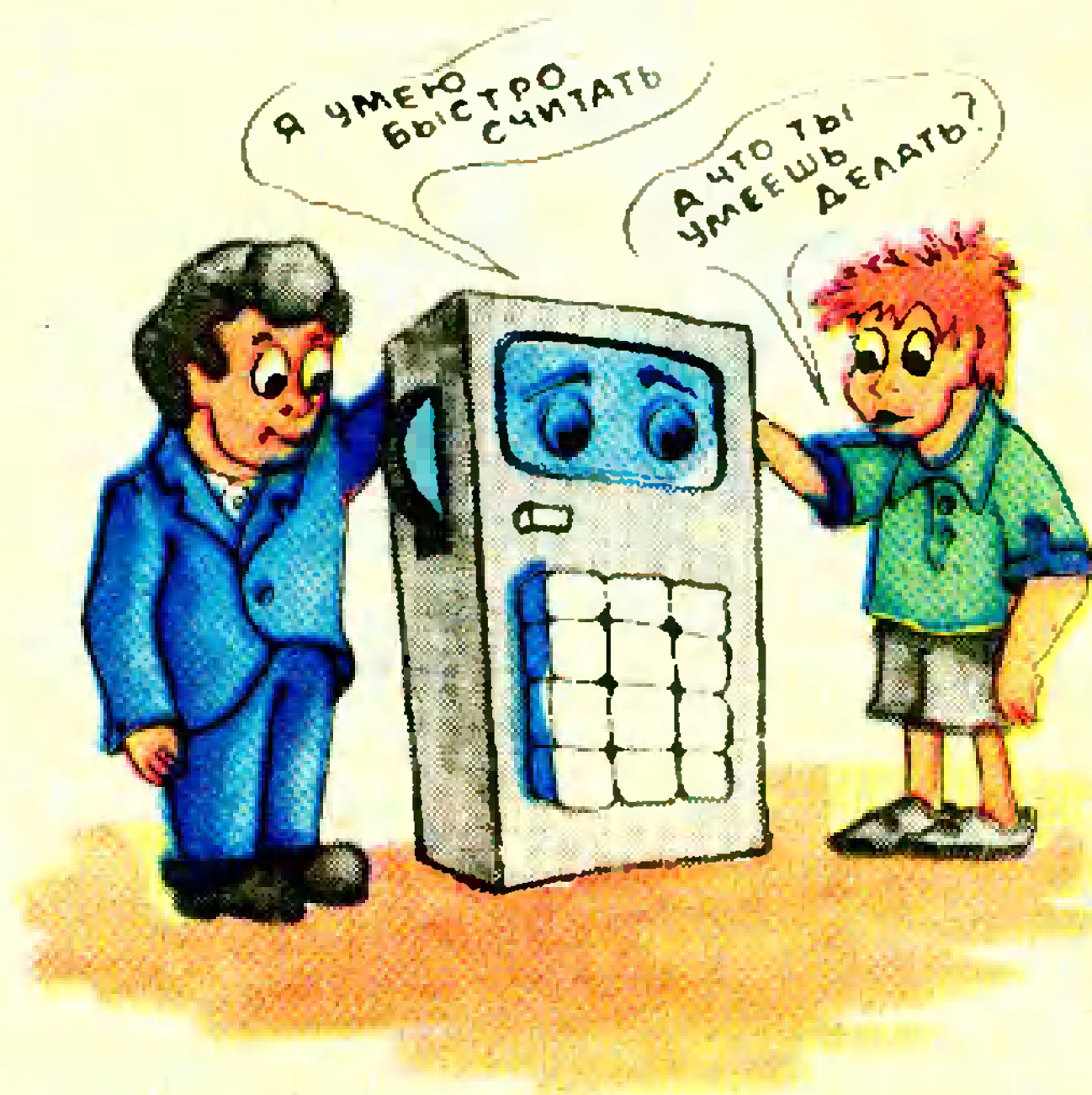
ресовался всем на свете и даже тем, чем ему не надо было интересоваться.

— А разве есть такие вещи, которыми нельзя интересоваться? — тут же спросил Любознашкин.

А Хвастунишкин очень любил хвастаться и говорил, что он всё умеет, хотя, конечно, умел не все...»

Сюжет сказки построен так, что содержит в себе структуру обучения, основанную на познании все более расширяющихся возможностей использования компьютеров в практической деятельности. Сказка разбита на части, соответствующие урокам первого полугодия. Занятия проводятся в виде игры, дети разбиваются на группы Любознашкиных и Хвастунишкиных: одни задают вопросы, а другие на них отвечают. Это способствует тому, что дети стремятся самостоятельно узнать как можно больше из мира компьютеров, о их применении на работе у родителей, собрать газетные и журнальные вырезки о ПК.

«...Друзья вошли в вестибюль компьютерной школы и увидели маленький компьютер.



* Сказка «Компьютерная школа» написана Я. М. Марголисом и его шестилетней дочерью Аней.

— Как тебя зовут? — спросил Любознашкин.
— Калькулятор, — гордо ответил малыш. — Или сокращенно — «Кальк».

— А кто здесь учится? — поинтересовался Любознашкин.

— Здесь учатся разные компьютеры, которые уже много чего умеют делать.

— А что ты умеешь делать? — спросил Хвастунишкин.

— Я умею быстро считать.

— Ну, сколько будет 3 прибавить 2?

Кальк тут же нажал на цифру три, потом на клавишу плюс (прибавить), затем на цифру два, наконец на клавишу равняется и у него в глазах загорелась цифра пять.

— Ох, уж невидаль какая, — сказал со скукой Хвастунишкин. — Я этот примерчик сделаю еще быстрее.

— Этот, может, и быстрее, — ответил Кальк. — А вот, например, к $37 + 68$, сколько будет?

Друзья переглянулись и задумались.

— Вот тут-то при работе с большими числами я вам и помогу.

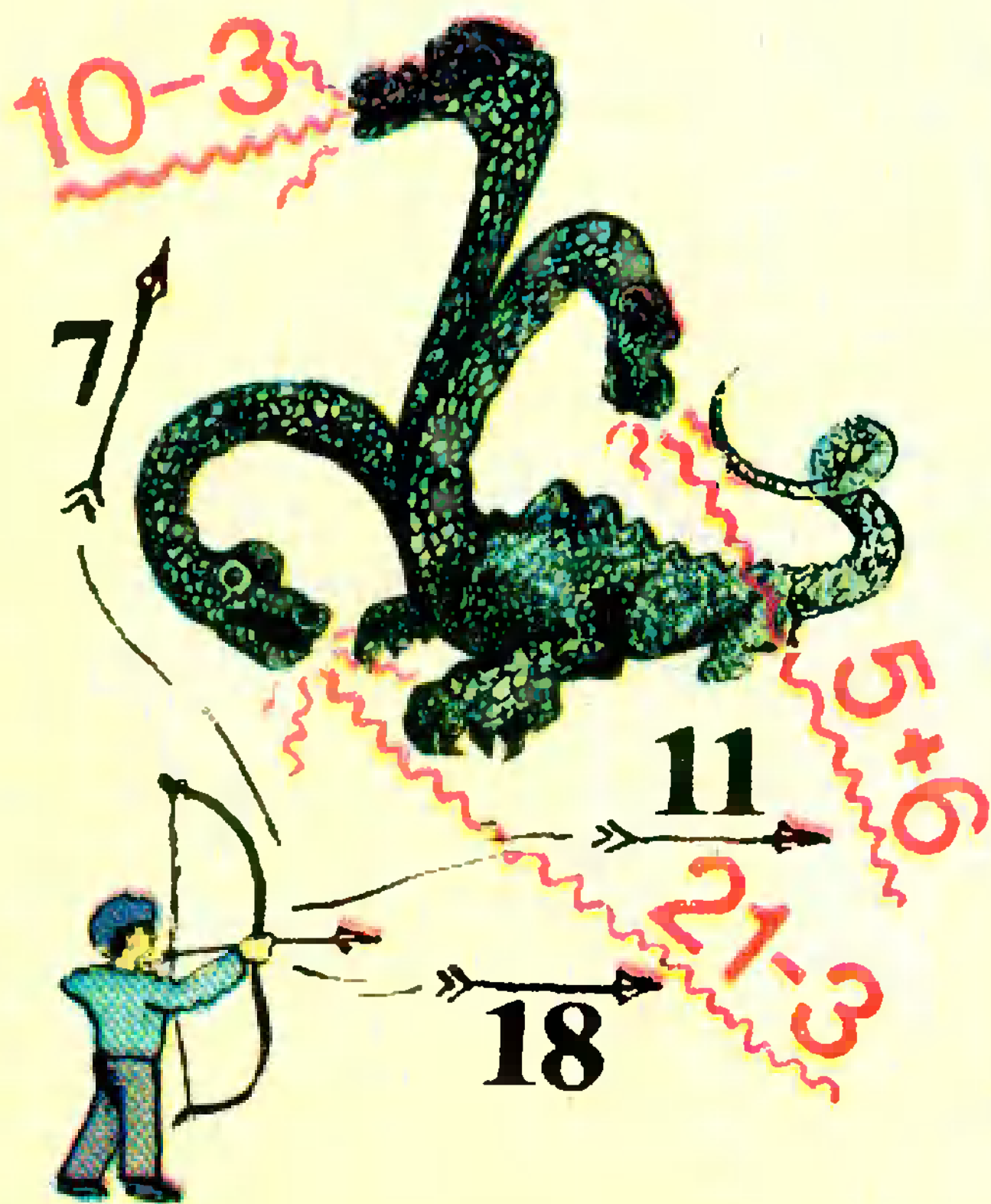
86 Он нажал на клавиши, и в его глазах загорелась цифра 105!

— Вот это да! — с восторгом сказал Любознашкин.

— Неплохо, — подтвердил Хвастунишкин...

Знакомство с миром компьютеров начинается с калькулятора. Дети учатся выполнять простейшие вычисления с целыми числами. В процессе обучения необходимо подвести детей к выводу о том, что возможности калькулятора при обработке информации ограничены. И не надо бояться того, что шестилетние дети разучатся считать: такие занятия, напротив, способствуют более быстрому усвоению детьми арифметических действий на традиционных уроках математики.

«...Любознашкин посмотрел на стену и увидел картину «Бой отличника Васи Петрова со Змеем Горынычем».



— А рисовать вы умеете? — спросил Любознашкин.

— Нет, не умею. Я только умею считать, а рисует у нас «Корвет». Он учится во II классе. Поднимитесь на второй этаж.

И друзья пошли дальше. «Корвет» занимал целый класс, и у него было много дисплеев.

— Вот это да, — сказал Хвастунишкин, — дисплеев — как у Змея Горыныча голов.

— Интересно, — сказал Любознашкин, — как на этих экранах рисуют картинки? На всех экранах по кусочку картины, что ли?

— Нет, — сказал «Корвет». — На одном экране одна маленькая картинка. Например, что вы хотите нарисовать?

— Мы хотим нарисовать таких же маленьких и смешных человечков, как мы сами.

— Пожалуйста, — сказал «Корвет» и нарисовал двух смешных человечков, под ними было написано: «Любознашкин и Хвастунишкин».

— А можно взять наши портреты на память? — спросил Хвастунишкин.

— Конечно, можно, — ответил «Корвет». — Сейчас я распечатаю вам их на бумаге. Это можно сделать с помощью устройства, которое по-русски называется «печать», а по-английски «принтер».

«Корвет» нажал на клавишу, и на бумаге распечатались смешные рожицы Хвастунишкина и Любознашкина.



Герои нашей сказки посмотрели на свои портреты.

— Но ведь у меня голубые глаза и рыжие волосы, — закричал Хвастунишкин. — А здесь только черный и белый!

— Если вам надо, чтобы картинки были цветные, идите к моему другу «Агату» в III класс. Он вам нарисует цветные картинки на своем дисплее».

Строя концепцию обучения шестилеток, мы исходим из того, что наиболее естественным, доступным и интересным практически всем детям на начальном этапе обучения является процесс рисования. Мнение современных ученых: работа пальцами развивает речь. В рисунке повествовать значительно

проще. Рисуя, ребенок отражает и упорядочивает свои знания о мире. Осознает себя в нем. Предварительно изобразив событие, ему легче рассказать о нем. Вот почему, по мнению специалистов, рисовать так же необходимо, как и разговаривать.

Поэтому после общего знакомства с калькуляторами первой самостоятельной творческой работой для детей является рисование на экране дисплея. Знакомство с графическим редактором — это не только рисование, но и познание формы различных геометрических фигур, их взаимного сопряжения, компоновки, а также развитие пространственного и конструкторского мышления.

Уже на этом начальном этапе обучения можно проводить работу, имеющую практическую значимость.

Творческое задание «Нарисуй самое красивое и самое некрасивое» позволяет выявить понимание детьми красоты. Задания типа «Чей узор лучше?», «Что такое лето?» способствуют не только художественному развитию детей, но и умственному. В этих случаях более подходит коллективная форма работы.

Дети любят всё, что сделано собственными руками. В процессе работы с графическим редактором можно прочесть с детьми их любимые книжки и подготовить к ним иллюстрации. Можно сделать книжку, состоящую из изображений животных или цветов. Важным моментом сравнения своих работ с чужими является «выставка». Здесь утверждается право каждого на индивидуальность и никто не подвергается критике.

С большим желанием дети работают на занятиях по созданию аппликации. Симметрия, ритм вызывают у них желание воспроизвести увиденное. Но даже на самом начальном этапе надо ставить перед ребенком условия выполнения задания не копированием образца, а его видоизменением. На занятиях изготовления аппликации дети учатся планировать свою работу, намечать последовательность выполнения операций, общаться друг с другом, что стимулирует развитие творчества ребят.

Примером коллективной работы также может служить проектирование на компьютере и дальнейшее строительство из кубиков детского городка.

На одном из занятий дети разбиваются на группы, в каждую из которых входят архитекторы (проектирующие город на компьютере), строители (собирающие город из кубиков), водители (доставляющие соответствующий материал) и т. д. Каждый из детей выбирает наиболее интересный для себя вид деятельности.

Свобода в выборе практической деятельности и способов работы, в чередовании дел является одним из важных элементов концепции обучения шестилеток. Но представленная ребенку свобода не только не исключает, а, наоборот, предполагает ненавязчивую, умную, доброжелательную помощь взрослых. Нужно не только показывать, а тесно сотрудничать с ребенком в процессе работы.

Такие виды работ позволяют реализовать схему: человек — компьютер — человек, которая значима с самого раннего возраста. Целью этой схемы обучения является формирование у ребенка представлений о том, что компьютер является не целью, а промежуточным звеном, средством, инструментом практической деятельности.

«Хвастунишкин и Любознашкин пошли к «Агату». Он встретил их очень приветливо и даже написал на экране дисплея приветствие «Здравствуйте, ребята!» красными буквами. Наши герои тоже поздоровались, и Любознашкин спросил: «Агат», а сколько цветов вы можете вывести на свой экран?»

— Восемь, — ответил «Агат» и перечислил: белый, черный, красный, желтый, зеленый, голубой, синий, фиолетовый.

— А где оранжевый цвет? — спросил Любознашкин. — Я так его люблю. Моя любимая песенка «Оранжевое море, оранжевое небо, оранжевая мама, оранжевый верблюд».

— К сожалению, у меня нет оранжевого цвета, — ответил «Агат».

— А как вы все цвета запоминаете? — спросил Любознашкин.

— У меня есть память, и с ее помощью я запоминаю разную информацию, например последовательность цветов в спектре.

— А что такое спектр? — спросил Любознашкин.

— Спектр — это радуга на небе, которая бывает после дождя, — ответил «Агат». — И в этой радуге есть все семь цветов: красный, оранжевый, желтый, зеленый, голубой, синий, фиолетовый.

— А как же вы их все запоминаете? — перебил очередной вопрос своего друга Хвастунишкин.

— У меня в памяти записана поговорка: «каждый охотник желает знать где сидят фазаны», — ответил «Агат».

— А причем тут Фазаны, — спросил Хвастунишкин.

— Вы заметили, — ответил «Агат», — что эти слова начинаются с тех же букв, как и цвета: красный, оранжевый, желтый, зеленый, голубой, синий, фиолетовый.

— А нам можно запомнить эту поговорку? — спросил вежливый Любознашкин.

— Конечно, можно, — ответил «Агат».

— Ох, как здорово! — сказал Любознашкин. — Прямо петь хочется.

— Я пою не очень хорошо, — ответил «Агат», — но у меня есть друг — КУВТ «Ямаха». Он родился в Японии и умеет воспроизводить ту музыку, которую вы ему напишите на экране дисплея.

— А он может воспроизвести мелодию из мультфильма «Ну, погоди!»? — спросил Хвастунишкин.

— Конечно, может!

— Ох, как здорово! — сказал Любознашкин, — Прямо петь хочется!

— Ну что ты все заладил: «Петь хочется, петь хочется!», пойдем к «Ямахе», там и споем, — пробурчал Хвастунишкин...

Изучая цветовую гамму дисплея, дети знакомятся с последовательностью расположения цветов в спектре. На наш взгляд, при обучении шестилеток наличие цветowych и музыкальных возможностей компьютера является необходимым элементом.

«... — А как нам пройти к «Ямахе»? — спросил Хвастунишкин.

— У нас есть класс, где работают иностранные компьютеры: американские — *IBM PC* и *Apple*, болгарские — «Правец-8» и «Правец-16», японские — «Ямахи» и другие компьютеры, их очень много, — ответил программист.

Этот иностранный класс находится на третьем этаже нашей школы.

И друзья побежали в иностранный класс.

По дороге Хвастунишкин, запыхавшись, говорил Любознашкину:

— *IBM* — это название фирмы, которая сделала этот компьютер, *Apple* — по-английски «яблоко», а яблоко — это плод познания, кто откусит от него, тот все узнает, так написано в самой первой книге — Библии, Правец — город в Болгарии, где делают эти компьютеры, а «Ямаха» — название знаменитой японской фирмы, выпускающей школьные компьютеры.

— Эта японская фирма выпускает не только школьные компьютеры, — ответил любознательный Любознашкин, — но и музыкальные инструменты.

— А я знаю компьютер, который исполняет музыку лучше, чем оркестр, — похвастался Хвастунишкин.

— Как играет оркестр мне понятно, — ответил Любознашкин, — а вот как исполняет музыку компьютер? Ведь в нем нет музыкальных инструментов.

Хвастунишкин задумался и сказал:

— А действительно, как же он играет, ведь в компьютер не засунешь пианино и даже маленькую скрипку?

— А в компьютере есть устройство, позволяющее имитировать различные музыкальные инструменты, — ответил радостный Любознашкин.

— Неужели компьютер понимает ноты? — настаивал Хвастунишкин.

— Конечно, — ответил Любознашкин, — есть специальная программа, позволяющая набирать на экране дисплея прямо на нотном стане разные ноты и потом прослушивать их.

— Но ведь так можно сочинять музыку, не владея ни одним музыкальным инструментом, а только набирая музыкальные фрагменты на дисплее и прослушивая их, — удивился Хвастунишкин.

— Конечно, если у тебя есть природный слух, — спокойно ответил умный Любознашкин, хотя был удивлен не меньше своего друга.

— А без слуха нельзя? — с грустью спросил Хвастунишкин.

— Без слуха можно, без компьютера нельзя, — ответил Любознашкин.

Второй большой раздел связан с изучением возможностей музыкального редактора. Для большинства детей — это первоначальное знакомство с нотной грамотой. Основным достоинством компьютерного освоения нотной грамоты является наличие обратной связи, когда ребенок, набрав музыкальный фрагмент, может его тут же прослушать и внести исправления, которые он считает необходимыми. После изучения музыкального редактора и знакомства с нотной грамотой ребенок может набирать собственные музыкальные фрагменты и получать распечатку нот.

Опыт показал, насколько полезно для общего развития детей вовлекать их в самостоятельную деятельность, воспитывать творческое отношение к музыке. Даже дети, не знакомые с нотной грамотой, но обладающие природным музыкальным слухом, могут заниматься сочинением собственной мелодии на компьютере.

«— Ну, хорошо. Допустим я написал свою мелодию. А как мне подписать ноты? — спросил Хвастунишкин.

— Подпиши так, — ответил Любознашкин, — композитор Хвастунишкин!

— А разве компьютер умеет писать? — спросил Хвастунишкин.

— Конечно, у меня есть специальная программа — текстовый редактор, которая позволяет писать на экране дисплея тексты, исправлять их, а затем распечатывать на принтере.

— «Ямаха», — спросил Хвастунишкин, — давайте посмотрим на ваших телевизорах мультики.

— Нет, — ответила «Ямаха», — это не телевизоры, а дисплеи. С них можно считывать цифры и буквы и писать на них слова и цифры.

— А чем писать, шариковой ручкой? — спросил Любознашкин.

— Нет, пишут с помощью клавиатуры. Видите, она рядом, — и очень похожа на пишущую машинку, — ответила «Ямаха».

— А, понятно, — ответил Хвастунишкин. — Чтобы написать на экране дисплея слово МАМА, надо нажать на клавиатуре буквы М, А, М, А.

— Правильно, — ответила «Ямаха». — Включайте русский шрифт и набирайте по буквам: композитор Хвастунишкин».

Текстовый редактор — один из обязательных элементов обучения шестилеток. Изучая текстовый редактор, дети учатся писать грамотно, не акцентируя внимания на каллиграфии. Здесь очень важна роль учителя, так как учащиеся допускают при написании очень много ошибок. Практическая направленность нашей концепции реализуется на этом этапе обучения путем создания классного журнала, в котором собраны

рассказы и стихи детей, подготовленные и распечатанные с помощью компьютера.

Главным в обучении шестилеток является не компьютер, а ориентация на развитие творческих способностей ребенка, самостоятельность в выполнении творческих работ.

Второе полугодие посвящено выполнению творческих работ на компьютере. Ребята самостоятельно выбирают темы и форму работы, ее объем. Это может быть рассказ или стихи на выбранную тему, рисование картинки и набор музыкального фрагмента, иллюстрирующего этот рассказ.

Работа может быть индивидуальная или групповая. В группе дети делятся по интересам: одним нравится рисовать, другим — сочинять рассказы, стихи или музыку, третьим — готовить тексты или ноты на дисплее, кому-то — заниматься распечаткой их на принтер (с помощью учителя), а кому-то переплетать полученные с принтера распечатки и делать из них компьютерный журнал класса.

И пусть такие работы еще не совершенны, но, главное, дети творят, и творят самостоятельно. В этом процессе главное, что каждый ребенок может выразить себя, раскрыть свои возможности. Такие занятия, конечно, требуют подготовки детей дома.

Чрезвычайно важным элементом обучения являются итоговые занятия «Семья за компьютером», на которых дети показывают своим родителям то, чему они научились за

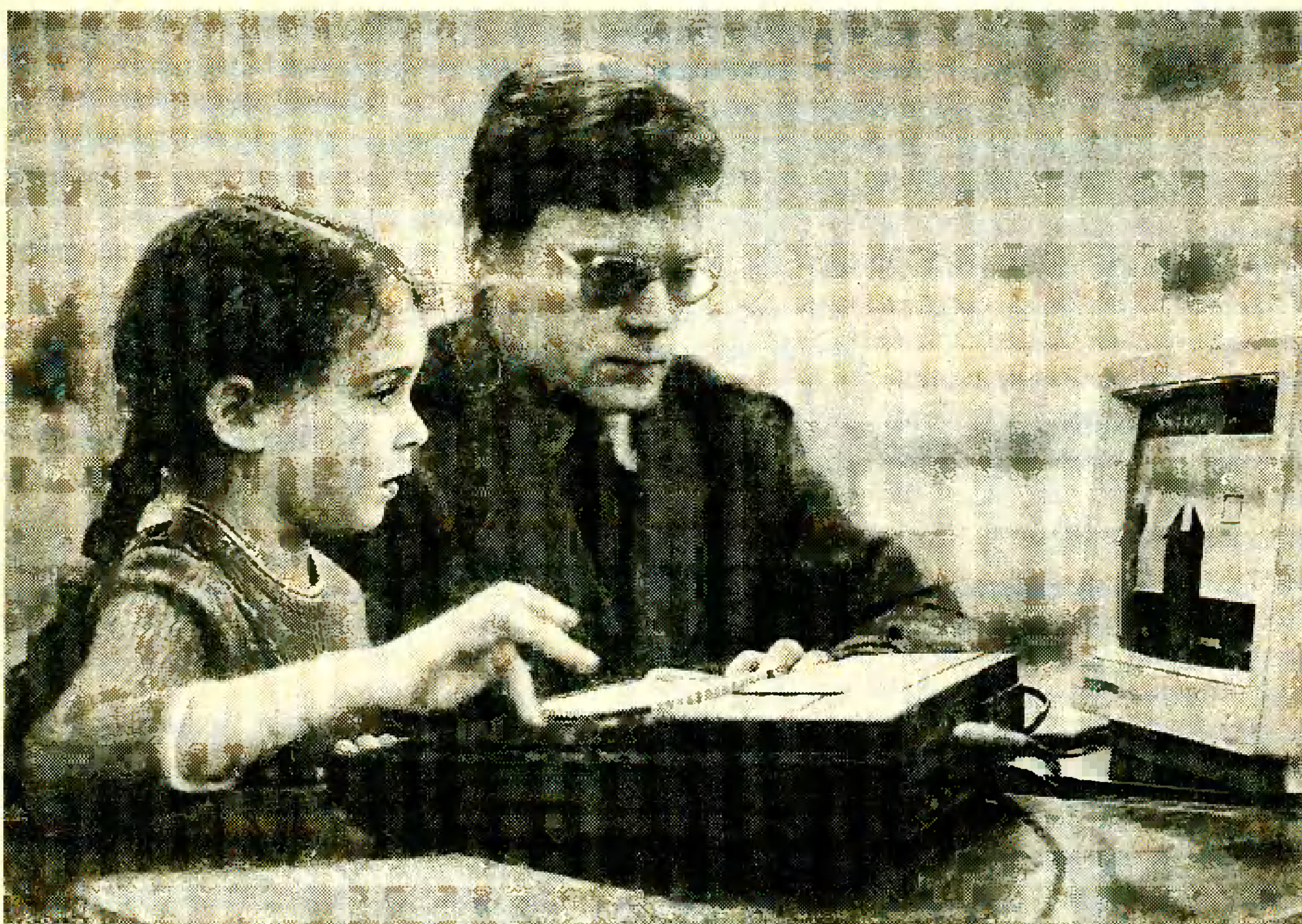
год, что они сочинили и выполнили своими руками, работая на компьютере. Эти занятия важны не только для детей, но и для родителей, многие из которых на этих занятиях получают первые представления о компьютерах. Парадоксальность этой педагогической ситуации в том, что дети шестилетнего возраста обучают своих родителей, дедушек и бабушек основам компьютерной грамотности. Это имеет большой педагогический эффект для ребенка, поскольку он осознает свою значимость.

Как показывает наш опыт, целесообразно, чтобы такие занятия, как «Семья за компьютером», носили регулярный, а не разовый характер. Компьютер в данной ситуации выступает связующим звеном в общении между родителем и ребенком.

Приведем примерный тематический учебный план, который может служить основой для проведения занятий с шестилетками.

Примерный тематический учебный план эксперимента по раннему обучению информатике шестилеток в средней школе № 63 г. Самары: Информатика-01 (И-01) 1 ч в неделю, 32 ч в год. Класс разбит на две подгруппы по 10 учеников в каждой, занятия на КУВТ «Ямаха».

Родительское собрание: цели и задачи раннего обучения информатике. Медицинские аспекты обучения с помощью компьютера. Экскурсия родителей в дисплейный класс и работа с игровой программой на КУВТ «Ямаха».



I класс

1. Знакомство с компьютером — 5 ч
- 1.1. Компьютеры и их возможности: фрагмент сказки для детей и родителей «Компьютерная школа». Экскурсия в дисплейный класс — 1 ч
- 1.2. Инструктаж по технике безопасности. Включение и выключение компьютера. Демонстрационная игра — 1 ч
- 1.3. Управление курсором: игра «Мудрый кролик» — 1 ч
- 1.4. Работа компьютера в режиме калькулятора — 2 ч
2. Компьютер — инструмент практической деятельности — 14 ч
- 2.1. Графический редактор: изучение назначения и возможностей — 2 ч
- 2.2. Работа по созданию рисунка с помощью графического редактора для: лепки из пластилина; вырезания аппликации; оформления классной стенной газеты и т. д. — 3 ч
- 2.3. Изучение нотной грамоты, возможностей и назначения музыкального редактора — 2 ч
- 2.4. Работа по созданию музыкального фрагмента к утреннику — 2 ч
- 2.5. Изучение назначения и возможностей текстового редактора — 2 ч
- 2.6. Работа по подготовке собственных рассказов и стихов с помощью компьютера для классной стенной газеты — 3 ч
3. Компьютер — средство развития

- творческих способностей ребенка — 11 ч
- Творческая работа на собственную тему по созданию с помощью компьютера рассказов, стихов с иллюстрациями в виде компьютерных картинок и музыкальных фрагментов.
- Изготовление из кубиков спроектированного на компьютере городка; лепка из пластилина фигур, полученных на компьютере; вырезание и наклеивание аппликации, нарисованной на дисплее и распечатанной на принтере; подготовка классной стенной газеты из работ, выполненных на компьютере.
4. Итоговые занятия «Семья за компьютером» — 2 ч

Отдельные фрагменты приведенной концепции обучения малышей творчеству через компьютер применяются и за рубежом. Так, наши немецкие коллеги из школы Karl-Ernst-Osthaus-Schule der Stadt Hagen сочиняют с детьми стихи, рисуют к ним иллюстрации в классном журнале [5]. Американские преподаватели из школы Марк Уэст, занимающиеся по методике педагога и поэтессы Морин Хэрли, делают упор на развитие образного и ассоциативного мышления при сочинении стихов.

В заключение приведем творческие работы, подготовленные на компьютере нашими шестилетками, немецкими и американскими детьми в начальной школе.

Весенние деньки

Пришла весна-красавица.
Красавица весна.
И детям очень нравится
По лужам погулять.

*Таня Горбунова,
Ири Рудометова*

Вот пришла весна

Мы очень рады,
И снег растаял, и течет вода,
И солнце ярко светит, и летят
Грачи, и сосульки с крыши каплют,
И капельки от них падают за
Шиворот!

*Катя Стрельцова,
Лена Кащеева*

Стихи, написанные учащимися
Karl-Ernst-Osthaus-Schule der
Stadt Hagen. Перевод стихов
В. Боголюбовой.

Красное яблоко

Человек гулял осенью,
А на дереве висело яблоко,
Человек сказал радостно:
— Ой, что я вижу,—
Чудесное яблоко.
Он тянулся, он вытягивался,
Но не мог его достать.
Тут человек сказал: «Я все же
достану тебя» —
И до тех пор тряс дерево,
Пока яблоко не упало.
С радостью он поднял его
И сразу съел.

Езика Ирш

Стихи детей школы Марк
Уэст (США). Перевод стихов
О. Атбашьяна.

Прилипла

Я прилипла!
Я не могу оторваться.
Я слышу — щебечут птицы.
Я прилипла!
Я слышу, как дышит лес
Как мне туда попасть?
Я здесь. Я прилипла!
Ветер — и совы уснули.
Я пробую вырваться снова,
И вот я свободна.
Лечу на небо. Я птица.
Плещу на луне и кричу:
Я — свободна!

Карли О'Коннор

Л и т е р а т у р а

1. Новоселова С. Проблемы информатизации дошкольного образования // ИНФО. 1990. № 2.
2. Иорки Т. ЭВМ в детском саду // Перспективы: Вопросы образования. ЮНЕСКО. 1987. № 4.
3. Хантер Б. Мои ученики работают на компьютерах.— М.: Просвещение, 1989.

4. Марголис Я. М., Иванов А. М., Баранкина Э. С. Содержание и методы непрерывного обучения информатике в начальной и средней школе // ИНФО. 1991. № 1.
5. Wiederhold K., Mitzlaff H. Computereinsatz in der Grundschiule? // Bildung und Computer, Hagen und Dortmund. 1989. Dezember. Heft 2.

И. БЕЛАВИНА

канд. психол. наук

факультет психологии МГУ

Психологические последствия компьютеризации детской игры

Психологические последствия компьютеризации детской игры — проблема, появившаяся относительно недавно в связи с повсеместным распространением персональных компьютеров и игровых компьютерных систем. Исторически сложилось так, что психологические последствия компьютеризации как научная проблема заявили себя впервые, когда возникла новая информационная технология в трудовой, научной деятельности, в обучении [1, 2]. Оказалось, что человек не готов эффективно, быстро и безболезненно перейти на новую ступень своего развития, его опыт и знания не обладают той степенью обобщенности и гибкости, которая бы помогла естественным образом включить новые информационные технологии.

Проблема психологических последствий компьютеризации, возникшая как проблема социальной, трудовой и общей психологии, в последнее время выделена в возрастной психологии в связи с распространением компьютерных игр среди детей самого разного возраста.

Каково психологическое воздействие компьютера на ребенка? Этот общий и важный вопрос следует конкретизировать в зависимости от той объективной реальности, которая доступна исследователю. Известный подход С. Пейперта [3] связан с обучением детей школьного возраста формальным методам решения задач. В нашем исследовании объектом анализа была игра дошкольника в компьютерно-игровом комплексе (КИК) в детском саду [4].

Анализ психического развития как стадийного позволяет оценить фактор воздействия компьютера с точки зрения общего и специфического содержания, привносимого в интеллектуальную и личностную сферу развития ребенка.

Компьютеризированная игра по своей организации не имеет аналогов в повседневной жизни ребенка, поэтому овладение способами взаимодействия с компьютером определяется во многом динамикой протекания различных психических процессов, степенью их зрелости и структурированности.

Рассмотрим отдельно эти два направления.

Как уже отмечалось, объектом изучения являлась игра ребенка с компьютером в усло-

виях КИК, которая оценивалась нами с более общих позиций развития творчества. Творческими составляющими любой деятельности, в том числе и детской игры, являются процессы целеобразования. Игровая деятельность характеризуется строением, мотивированностью участников игры, наличием правил игры, способов выполнения действий, игровой моделью.

В условиях компьютеризации игровой деятельности существенные изменения, отражающие закономерности овладения средством деятельности, каковым должен явиться компьютер, происходят прежде всего в познавательной сфере ребенка. Становясь подлинным средством, компьютер позволяет «уравновесить» неравномерность динамики развития психических процессов (восприятия, образного, вербального, логического мышления) — принцип дополнительности. С одной стороны, компьютер является для ребенка сложным объектом познания, овладение которым предъявляет более высокие требования к организации деятельности и ее структурированности, т. е. определенной зрелости интеллектуальных структур (цель — средство — результат). С другой, становясь средством, компьютер позволяет ребенку реализовать такие потенциальные возможности, которые в традиционных условиях не могут проявиться, например, из-за несформированности графического образа или стереотипного способа реализации графического образа, что убедительно показано в работе В. С. Мухиной [5].

Персональные компьютеры по распространенным на сегодня способам взаимодействия с ними предопределяют индивидуализированные формы организации детской игры. В исследовании Е. Е. Лысенко обосновывается тезис о видоизменении стадии игры ребенка с предметом. В чем новизна качества, привносимого компьютером? Во-первых, компьютер переводит на более обобщенный уровень само действие с предметом: оно дополняется внутренним образом, представлением об условности, нереальности перемещения игровых элементов, например, в игре с программой «Конструктор». Это означает, что освоение компьютера как средства требует полноценности уже сформированных у ребенка способов действия с

реальными предметами на уровне наглядно-практического действия, т. е. соответствия требованиям формирования обобщенного способа действия. При условии выработки у ребенка на стадии игры с предметом некоего стереотипа (шаблона) компьютер ставит перед ребенком сверхсложную задачу, решить которую без помощи взрослого он не в состоянии. Следовательно, на этапе овладения компьютером со всей очевидностью проявляются огрехи педагогического воспитания и обучения, которые были совершены на стадии овладения способом действия с реальным предметом или предметом-заменителем. Можно предположить и некоторый эмпирический опыт, позволяющий утверждать, что компьютер может быть использован в игре ребенка с предметом на соответствующей этой игре возрастной ступени, начиная с 3 лет. Компьютер расширяет вариативность способов преобразования игровых элементов, способов их включения в игру, т. е. создаются условия для полноценного формирования способов действия. В старшем дошкольном возрасте, когда преобладает сюжетно-ролевая игра, для того чтобы взаимодействие ребенка с компьютером переросло в игру, требуется перестройка уже сложившихся стереотипизированных способов действия.

Каким образом развивается игра ребенка с компьютером? Наблюдение за детьми старшего дошкольного возраста (начиная с 5 лет) показало, что следует выделять несколько этапов ее развития. Во-первых, дети по-разному относятся к компьютерным программам: с одними программами они охотно играют, эти программы выделяются ими как любимые («Конструктор», «Золотой ключик», «Войди в замок»), другие программы ребенок быстро забывает, не столь часто просит для игры («Сложи картинку», «Город», «Гараж», «Лиса и цыплята»). После 3—4 предъявлений одной программы дети в основном перестают ею интересоваться. Их привлекает новизна программы, ее занимательность, красочность, музыкальное оформление. Обращает внимание факт достаточно быстрого угасания желания ребенка играть в одну и ту же игру, несмотря на то что дети до конца не исчерпывают возможности ряда программ, например «Сложи картинку». Взаимодействие с компьютером в тот короткий период, который регулируется медико-эргономическими требованиями, должно быть определенным образом организовано. В этом состоит отличие развития компьютеризированной игры от традиционной игры ребенка с предметом. Если в компьютерном зале создаются условия, близкие к естественным фор-

мам развития игры, то это способствует стихийной форме овладения компьютером как средством игровой деятельности. От степени развитости познавательной активности зависит и эффективность овладения этим новым для ребенка орудием познания мира.

Компьютеризированная игра не может инициироваться ребенком самостоятельно, без предварительного выполнения определенных вспомогательных действий взрослого по загрузке, перезагрузке программ, подготовке всей системы в целом. Компьютеризированная игровая деятельность имеет строгую последовательность действий и пауз, в течение которых происходит произвольная смена форм активности ребенка. Внешние условия могут создавать помехи в развитии детской игры и приводить к ее деструктурированию и разрушению.

При естественной форме организации детской игры (свободная игра) наблюдаются несколько типов поведения ребенка. Наименее продуктивным оказывается поведение, двигательным моментом которого является любопытство как реакция на новый предмет. Ребенок играет с компьютером, ориентируясь на простейшие способы взаимодействия с ним, используя шаблонные приемы типа «нажал кнопку и увидел что-то». При недостаточной степени развитости познавательной активности любопытство не переходит в любознательность, и при появлении сбоя или иных помех игра заходит в тупик и прекращается. Потребность в новизне быстро удовлетворяется, и ребенок легко переходит к другому компьютеру, к другой программе. Действия наугад, бессистемные, не способствуют в таком случае образованию обобщенного способа действия и не формируют познавательную потребность — источник дальнейшего развития детской игры. Ребенок легко переключается на иные традиционные формы игры. Переноса каких-либо компьютерных знаний в игровую деятельность (без компьютера) не происходит.

В зависимости от степени сформированности целевых компонентов, уровня развития познавательной активности, а также от содержания игровой деятельности отдельные дети возобновляют прерванный внешними обстоятельствами (сбой, перезагрузка программы) игровой процесс, сохраняя общую цель, но, изменяя форму воплощения замысла, например, в программе «Конструктор» вместо прежнего рисунка создается новый (более продуктивный тип поведения).

Сложность и новизна компьютера обуславливают необходимость учета этапа его освоения, на котором особую значимость приобретает форма организации детской иг-

ры (ее планирование, подготовка, вхождение в игру) и форма организации участия взрослого в ее осуществлении.

Важность этого показал анализ поведения детей старшего дошкольного возраста в компьютерном зале. Общеизвестно, что на этой возрастной ступени развития для детей большей привлекательностью обладают сюжетно-ролевые игры как форма коллективной игры. Несмотря на теоретические гипотезы о естественности перехода от игры с предметом к сюжетно-ролевой игре начиная с 5 лет, дети в компьютерном зале не организовывали совместную игровую деятельность. Не отмечалось случаев сюжетно-ролевых игр вне компьютерного зала (при условии стихийной формы организации компьютеризированной игры). Одновременно организовывались группы (2—3 ребенка), в случае когда у одного из детей прерывалась игра из-за неумения ее продолжить. У детей не возникает потребность в совместной деятельности, вероятно, из-за того, что ребенок не исчерпал свою потребность в познании компьютерных возможностей. Объединение детей чаще всего носило неорганизованный характер, их действия не были структурированы единой целью или замыслом, у них не возникли правила совместной игры, не было ярко выраженных ролей. Дети объединялись лишь для того, чтобы показать, как овладеть тем или иным способом игры с компьютером. Можно было наблюдать, что совместные формы помощи более частое явление; дети с более обобщенным, сформированным способом действия, овладевшие навыками управления конкретной компьютерной программой, довольно охотно помогают другим. У этих детей также более развита потребность в оценке, в самостоятельных действиях и достигнутых результатах. Стихийно складывающаяся форма совместной деятельности, если она имеет место, может приобретать характер индивидуальной, когда более активный ребенок вытесняет менее активного, переходя, например, от совместного «рисования» с помощью программы «Конструктор» к индивидуальному продолжению работы с начатым изображением. В большинстве случаев дети не умеют играть коллективно на компьютере, несмотря на все попытки педагога спровоцировать подобную форму игры. Вероятно, потребность в совместной деятельности может сформироваться лишь при определенных психолого-педагогических формах организации сначала игры со взрослыми, затем, по мере приобретения ребенком навыков игры с компьютером, его индивидуальная игровая деятельность перерастет в совместную. Это не простой и не быстрый

путь, на котором роль умелого и вдумчивого взрослого важна и значительна. В компьютерной игре взрослый не только учит ребенка способам организации игры, но и стимулирует развитие любознательности, которая, как интеллектуальное чувство, питает познавательную активность, пробуждает волевые компоненты, структурирующие деятельность, пробуждает потребность в социальном одобрении окружающих продуктов игровой деятельности ребенка. Без участия взрослого ребенок не в состоянии овладеть таким новым игровым средством, так как у него нет в повседневной жизни ни опыта подобной игровой деятельности, ни его аналогов в повседневной жизни взрослых, перенять и подражать которым он бы мог.

За короткий период неорганизованного взаимодействия с компьютером без предварительной подготовки и настроя ребенок осваивает в основном принцип «кнопочной» игры, который в значительной мере ограничивает широту игровой деятельности и потенциально содержит в себе предпосылки для шаблонизации восприятия всего, что связано с компьютером.

Таким образом, в условиях дошкольного воспитания и дефицита времени, проводимого ребенком за компьютером, естественная, свободная форма организации игры на стадии освоения провоцирует формирование пассивного отношения ребенка к компьютеру (при условии, невысокой познавательной активности и любознательности), выработке шаблонных способов действия и создает эмоционально стрессогенные условия восприятия компьютера. Если же игра приобретает характер компьютеризированной деятельности, то компьютер способствует расширению содержания общих интеллектуальных навыков и умений, обогащению способов действия и стимулирует разрушение и преобразование шаблонизированных приемов игры.

Следующим новым моментом в игровой деятельности ребенка является формирование потребности в сотрудничестве со взрослым или другим, более развитым ребенком, свободно владеющим приемами игры с компьютером. Однако следует заметить, что потребность в сотрудничестве была зафиксирована как стабильное образование лишь при организационных и спланированных формах игровой деятельности под руководством взрослого. Потребность в сотрудничестве появляется при определенной степени интенсивности познавательной активности. Как уже было замечено, если движущей силой является любопытство (реакция на компьютер как на новый предмет), встречающиеся на пути развития игры препятствия

ребенком не преодолеваются, он легко отказывается от игры с компьютером. Обращаясь за помощью к другому, ребенок тем самым стремится сохранить и продолжить игру, даже если в данный момент имеющиеся у него знания и навыки не позволяют достигнуть поставленной им цели. Ребенок тем самым вовлекается в «зону ближайшего развития». В этом случае совместная деятельность носит иной характер, в отличие от стихийного проявляющейся. Ребенок активно следит за действиями взрослого, порой вмешивается в процесс взаимодействия с компьютером, когда это ему по силам. Это вмешательство всегда импульсивно, эмоционально и либо сопровождается репликами типа: «Теперь я сам», либо ребенок нетерпеливо отталкивает руку взрослого или другого ребенка, чтобы самому осуществить нужное действие.

Таким образом, с одной стороны, игры с компьютером требуют наличия определенного уровня познавательной активности и ее интенсивности, а с другой стороны, сам компьютер является мощнейшим источником развития как познавательной активности, так и других интеллектуальных чувств ребенка — любознательности, удовлетворенности результатом своей деятельности, а также волевых качеств, позволяющих сохранить и удержать процесс игры, даже в случае сильных внешних помех.

Новизна компьютера состоит также и в том, что взаимодействие с ним очень эмоционально. Это создает благоприятные условия для развития не только интеллектуальных компонентов деятельности, но и

определенных мотивационно-личностных образований. Игра с компьютером не насыщаема (при адекватных формах ее организации) благодаря включению в игровую деятельность все новых и новых программ, а также благодаря тому, что эти программы обладают разной степенью сложности.

Уже отмечалось, что ребенок может включаться в игру с разной степенью развития мотивации: от простого любопытства, любознательности до увлеченности и полного погружения. Наряду с этими мотивационными образованиями взаимодействие с компьютером окрашивается ярким эмоциональным чувством удивления, возникающего при реализации замысла, при получении правильного ответа. Бесспорно, что эмоции удивления в значительной мере прививаются путем воспитания и общения со взрослым (например, педагогом КИКа) и могут иметь разную глубину и продолжительность. При поверхностном вовлечении в игру, когда игра не захватывает воображение ребенка, ребенок очень небрежно выполняет игровые действия, отвлекается, шалит и легко переключается на другие предметы, особенно при появлении каких-либо сложностей или ошибочных действий, заводящих в тупик. В таких случаях познавательная активность практически не приобретает развивающего характера. На следующих уровнях активности, по мере возрастания вовлеченности в компьютеризированную игру, она приобретает характер деятельности, т. е. ошибки или иные помехи не разрушают процесс игры и не снижают уровень мотивационной составляющей.

Продолжение следует.

Вы нам писали

В № 2, 4 «ИНФО» за прошлый год читатели могли познакомиться с материалами новой рубрики «НИТ в дошкольном образовании». С волнением и надеждой ждали мы первых откликов.

«Я занимаюсь обучением будущих учителей начальных классов информатике. Хотелось бы установить КИК (компьютерно-игровой комплекс) в одном из детских садов и начать обучение детей 4-5-лет,— пишет Марголис Я. М. из г. Куйбышева.— Где можно приобрести программно-методическое обеспечение и КИК?».

«Где можно подробнее узнать о КИК, его структуре, техническом составе, возможности приобретения педвузом?»,— спраши-

вает преподаватель Армавирского пединститута В. В. Баем.

С такими же вопросами обратились к редакции директор УПК г. Оренбурга В. Я. Шпутин, КООП «Электроника» г. Пскова, методист Горметодкабинета г. Полевского и многие другие.

Мы попросили ответить на них Генерального директора Ассоциации «Компьютер и детство» В. Ю. Демьяненко:

Ассоциация «Компьютер и детство» разрабатывает на широкой междисциплинарной научной основе компьютерно-игровые комплексы и осуществляет их поставку, как говорится, «под ключ».

В состав комплекса входят информа-

ционно-вычислительная сеть на базе микро-ЭВМ, программное обеспечение и методические рекомендации для педагогов по использованию компьютерных игровых и обучающих программ.

Программное обеспечение КИК в качестве первого пакета содержит 35 программ компьютерных игр, разработанных программистами совместно с учеными — педагогами и психологами.

КИК «Электроник» поставляется в двух вариантах: в комплекте технических средств с пакетом прикладных программ и методическими рекомендациями для педагогов и в комплекте с пакетом прикладных программ на магнитных носителях. Дополнительно по заказу могут быть поставлены: комплект мебели для компьютерного зала (или чертежи мебели); оборудование для физкультурного и игрового залов. По договору с заказчиком предоставляются услуги по монтажу, пуско-наладочным работам и обучению работе на технических средствах КИК. Гарантийное и послегарантийное обслуживание технических средств КИК осуществляет производственное объединение «Электронсервис».

«...Будем объективны и посмотрим на дело реалистично: многие детские сады сегодня не имеют элементарного, а вы рассказываете о заоблачных высях...», — пишут заведующие детскими садами (фамилии мы решили не называть).

«...Думаю, что проблемы, поднятые Вами, вряд ли будут скоро решены».

«...«Компьютер в детском саду» — звучит, может быть, и заманчиво, но сколько у нас детей с нарушением зрения, осанки и пр.» Не увеличат ли НИТ эти горькие проценты?..»

Мы разделяем тревоги наших читателей. Действительно, многие детские учреждения не укомплектованы необходимыми игрушками, пособиями, оборудованием. Требуется радикально изменить положение дел с материальной базой учреждений начального звена народного образования. Много в этой области сегодня зависит от местных Советов — найти резервы, организовать на

предприятиях местной промышленности выпуск игрушек, пособий и оборудования для детских садов, обеспечить заявки и поставку средств воспитания дошкольников.

Современная, отвечающая психолого-педагогическим и медико-гигиеническим требованиям (см. «ИНФО» № 6, 1990 г.) материальная база — одно из обязательных условий для установки КИК.

В целях радикального изменения положения дел с оборудованием детских садов пособиями и игрушками, техническими и информационными средствами Гособразованием СССР создано малое государственное предприятие «Центр Дидактика». В 1991 г. планируется разработка системы развивающей предметной среды детского сада. К этой работе привлечены высококлассные специалисты в области дошкольной педагогики и психологии, медики, биологи, эргономисты, работающие в творческом сотрудничестве с Союзом дизайнеров СССР и Союзом театральных деятелей СССР. Надеемся, что такую же гражданскую позицию — «Помочь детству» — займут и предприятия, которые могут выпускать игрушки, пособия, мебель и оборудование (химики, от которых мы ждем новых красителей и материалов, инженеры-конструкторы и другие специалисты). С наименьшими надеждами мы обращаемся и в высокие государственные органы, чтобы призыв «Все лучшее — детям!» наконец был реализован не на словах, а на деле.

Очень надеемся и на помощь каждого из читателей. Ведь детям ждать некогда — они растут. А от того, с кем они общаются и чем занимаются, зависит, какими они вырастут.

Мы благодарим Вас, А. И. Демидова из г. Краснодара, Н. М. Котова из Ташкента, Н. Ф. Комарова из г. Нижнего Новгорода и многих-многих других за письма, телеграммы, звонки с добрыми напутствиями, пожеланиями, предложениями о помощи и сотрудничестве. Ищем возможность организовать встречу и дальнейшие творческие контакты.

Н. РЕУЦКАЯ

ВНИМАНИЮ РУКОВОДИТЕЛЕЙ

отделов народного образования, детских садов, детских домов и школ-интернатов, педагогических учебных заведений, а также предприятий, имеющих ведомственные детские школьные учреждения!

Предлагаем Вам новую информационную технологию разностороннего творческого развития детей младшего возраста — КИД-систему, в которой эффективно сочетаются новейшие педагогические принципы и технические достижения, развивающие и обучающие компьютерные игры и традиционные виды детской деятельности.

Применение системы КИД в дошкольном воспитании обеспечивает развитие умственных способностей, творческой инициативы детей, создает условия формирования через игру психологической готовности к использованию вычислительной техники в учебном процессе, быту и в будущей трудовой деятельности.

КИД является многокомпонентной гибко формируемой и развивающейся системой. В нее входят средства вычислительной техники (ВТ), игровые развивающие и обучающие программы с методическими рекомендациями, специальная мебель под компьютеры, игры и игрушки, спортивное оборудование и другие элементы детской предметной среды.

Средства вычислительной техники КИД-системы

Поставка средств вычислительной техники КИД может быть осуществлена в следующих вариантах:

1. Комплекс на отечественной ВТ (БК-0010.01, БК-0011):

рабочее место педагога на базе ПЭВМ БК-0011 с накопителем на гибких магнитных дисках, печатающим устройством, видеомониторами цветного и черно-белого изображения;

десять индивидуальных мест дошкольников на базе микроЭВМ БК-0010.01 с видеомониторами цветного изображения, объединенных с рабочим местом педагога в информационную сеть;

магнитофон «Электроника-302» в качестве резервного запоминающего устройства.

2. Комплекс на зарубежной ВТ:

рабочее место педагога на базе ПЭВМ типа IBM PC/AT или PS/2 (модель 30) с накопителями на жестком и гибких магнитных дисках, печатающим устройством, видеомонитором EGA или VGA;

три — пять индивидуальных мест дошкольника на базе ПЭВМ типа IBM PC/XT с накопителями на гибких магнитных дисках с видеомониторами цветного изображения, объединенных с рабочим местом педагога в информационную сеть.

3. Комплекс на аналогах IBM PC отечественного производства.

Данный комплекс по составу технических средств и их возможностям равноценен варианту зарубежной ВТ.

Более подробная информация будет выслана на ваш запрос. За справками обращаться по адресу: 116607, Москва, Мичуринский пр., д. 31, корп. 5. Ассоциация «Компьютер и детство». Телефоны: 931-44-55, 932-60-71.

О. КОЗЛОВ

канд. техн. наук

Поэтапная подготовка учителей-предметников к проведению занятий в кабинете информатики

Можно говорить о том, что первый этап внедрения вычислительной техники в школы завершается: кабинеты информатики функционируют во многих школах, подготовлены учителя информатики, у которых есть даже возможность выбирать учебники; активную позицию в вопросах компьютеризации образования заняли педагогические институты. Но реальная компьютеризация осуществится в школе только тогда, когда ее идеями проникнется большая часть учителей и проведение занятия по любому предмету в кабинете информатики станет нормальным явлением.

Для этого необходимо приобщить учителей к новому для них миру компьютеров, устранить психологический барьер их «боязни», т. е. грамотно организовать систему «обращения в компьютерную веру» всех учителей, уже работающих в школе (о тех, кто придет в ближайшие годы, позаботятся в педагогических институтах). Но в этом вопросе за последние годы допущен ряд ошибок. Главная из них заключается в том, что энтузиасты, взявшиеся за дело компьютеризации, решили «подтянуть» до своего уровня и всех остальных учителей. На курсах в ИУУ преподаются основы информатики, изучаются языки программирования, осваивается клавиатура компьютеров, делается многое другое, остается не ясно учителю физики или биологии одно: а как же это все применять на уроке? «Потеряли» принцип постепенности освоения нового и весьма сложного технического средства обучения, из всего многообразия проблем основной упор сделан на программирование, что и привело скорее к отрицательным, чем к положительным результатам.

Вниманию читателей предлагается вариант поэтапного подхода в приобщении учителей к компьютерной технологии обучения, причем каждый из учителей может остановиться на том уровне подготовки,

который считает для себя достаточным и посильным.

Что же включает в себя понятие «проведение урока с использованием компьютера»? На уроке информатики компьютер, как правило, объект изучения. На остальных уроках — техническое средство обучения, используемое для активизации учебного процесса.

Для того чтобы компьютер оправдал свое присутствие на уроке, необходимо провести большую подготовительную работу, включающую следующие основные этапы:

1. Составление сценария работы с компьютером.

На этом этапе в словесной форме, в виде таблиц или рисунков, задаются последовательность предъявляемого с экрана материала, варианты ответов и возможная реакция компьютера на них. Никаких специальных знаний по вычислительной технике для этого не требуется, а вот знать предмет и методику преподавания надо хорошо. Это — самая «наукоемкая» часть работ по подготовке компьютерного занятия. Именно здесь может раскрыться методическое мастерство одного учителя с тем, чтобы десятки и сотни коллег могли воспользоваться его опытом.

2. Реализация сценария на конкретном типе компьютера. Эта часть работы требует серьезной профессиональной подготовки прикладного программиста, которому неплохо иметь представление о психологических особенностях общения человека с компьютером. В последнее время появился ряд разработок, позволяющих преподавателям реализовывать свои сценарии программ с помощью удобных диалоговых средств, таких как Ирис, Астра, Астра-Микро, Рига и др. Однако профессиональное владение этими средствами, умелое использование графики, цветовых и звуковых эффектов возможно только при наличии глубоких знаний и навыков практической работы с компьютером.

Хорошая программа чаще всего является результатом коллективного труда педагогов, программистов и психологов. Каталоги разработанных обучающих программ издаются многими вузами и НИИ АПН СССР.

3. Подготовка методической стороны урока.

Обучающих программ по различным дисциплинам много; но как применить их на конкретном уроке? Здесь уже не нужны знания программиста, но в обязательном порядке необходимы глубокое знание предмета, методика проведения занятий. И опытный методист должен определить цель использования компьютера на уроке, заранее определить педагогический эффект, достигаемый за счет применения компьютера. Сюда можно отнести такие моменты, как:

изложение сложного материала с помощью программ-моделей и иллюстрирующих программ, в том числе игровых;

98 моделирование процессов и ситуаций для принятия учениками правильного решения;

контроль усвоения материала предыдущего и текущего уроков;

формирование практических навыков с помощью программ-тренажеров и т. п.

Следует отметить, что совсем не обязательно планировать использование компьютера все 45 мин урока. Для проведения контроля достаточно 10—15 мин, иллюстрации нового материала — 20—25 мин, но при проведении сложных расчетов, решении комплексных задач, связанных с перебором и выбором оптимальных решений, могут потребоваться и сдвоенные уроки.

В целом разработки опытных методистов, дополненные перечнями используемых программ, могут оказаться полезными для многих учителей, желающих использовать на своих уроках компьютеры. В ме-

тодических разработках необходимо в общих чертах сказать о характере информации, предъявляемой на дисплее, описать возможные действия учеников и определить пути выхода из «тупиковых» и ошибочных ситуаций.

4. Практическая работа с вычислительной техникой.

Итак, программы разработаны, методические разработки отпечатаны, планы написаны; теперь остается провести урок на самом высоком уровне. А кто будет включать компьютеры, загружать программы, отвечать на вопросы учеников и т. п.? Хорошо, конечно, если зав. кабинетом информатики поможет. А если он занят? Тогда необходимо делать все самому, т. е. иметь практические навыки работы с персональным компьютером, с его внешними устройствами. Особой сложности в этом нет, но учиться надо.

Опыт подготовки учителей и преподавателей вузов позволил систематизировать перечисленные выше этапы подготовки и проведения занятия с помощью компьютеров и оформить их в виде следующей таблицы. Предложенная терминология условна, однако она достаточно полно отражает уровень подготовки.

Ориентируясь на таблицу, факультеты и курсы ИУУ могут организовать подготовку учителей различных уровней квалификации, исходя из имеющейся в регионе техники, собственных возможностей, а также, учитывая пожелания учителей.

На уровень квалификации «Гость в кабинете информатики» можно подготовить учителей-предметников путем проведения открытых и показательных занятий. Для достижения квалификации «Арендатор кабинета информатики» нужно освоить включение и выключение класса, правила техники

Характеристика уровня подготовки учителя	Вид деятельности					Подготовка сценария
	Проведение урока в кабинете информатики	Умение работать с компьютером	Разработка плана и методики проведения урока	Разработка программных уроков		
				на языке программирования	с помощью АОС	
«Гость»	+					
«Арендатор»	+	+				
«Методист»	+		+			
«Практик»	+	+	+			
Разработчик сценариев						+
Прикладной программист		+		+		
Пользователь АОС		+			+	
«Универсал»	+	+	+	+	+	+

безопасности, работу с магнитными носителями, действия при сбоях программ, назначение клавиш компьютера и т. д.

Более высокие уровни подготовки требуют и более продолжительной учебы, но именно поэтапность может создать необходимую базу для повышения квалификации.

Очевидно, можно рассмотреть и другие сочетания навыков, но они будут касаться вопросов подготовки специалистов высокого уровня. Здесь мы готовы воспользоваться предложениями наших коллег из других городов.

Ленинградское МХП «Алмаз» предлагает

учебные компьютерные классы на базе персонального компьютера «Спектрум-64К» в различной комплектации.

«Спектрум-64К» — 8-разрядный ПК класса “ZX Spectrum” оригинальной разработки с микропроцессором Z 80 (3, 5 МГц):

объем памяти — 64 К байт;

80-клавишная двуязычная клавиатура профессионального типа (с английским и любым национальным алфавитом);

отображение информации на цветном или черно-белом мониторе;

полная программная совместимость с ПК “ZX Spectrum” английского производства;

широкий набор игрового и учебного программного обеспечения.

Учебный компьютерный класс — это:

до 32 ПК «Спектрум-64К», объединенных в локальную вычислительную сеть;

до 4 дисководов, матричный принтер с возможностью доступа с любого ПК

класса;

возможность контроля процесса обучения с рабочего места учителя;

программное обеспечение, дающее широкие возможности для организации учебного процесса в школе.

Компьютерные игровые автоматы «Спектрум» — это:

специальная модификация ПК «Спектрум-64К»;

объем памяти — 64 К байт;

отображение информации на цветном мониторе или телевизоре через специальный вход;

наличие разъема подключения джойстика;

загрузка с любого кассетного магнитофона (возможен вариант с картриджем);

возможность подключения 40-клавишной клавиатуры, превращающей игровой автомат в персональный компьютер;

увлекательное игровое программное обеспечение на кассетах или картриджах.

Ленинградское МХП «Алмаз» ищет дистрибуторов названной продукции.

198096, Ленинград, а/я 19. Тел. для связи в Ленинграде — 152-52-58.

ВНИМАНИЕ: конкурс студентов!

В 1991 г. Республиканский Центр новых информационных технологий в образовании Минобразования РСФСР при Омском государственном педагогическом институте в соответствии с планом работы секции подготовки педкадров общесоюзного НМС информатизации образования Гособразования СССР проводит **ПЕРВЫЙ КОНКУРС** работ студентов педагогических вузов по разработке программных средств (ПС).

Основная цель конкурса — активизация творчества будущих учителей по разработке и использованию программных средств и новых информационных технологий в сфере образования. На конкурс принимаются программные средства, ориентированные на учебный процесс и управление в средних школах, инструментальные и другие ПС, направленные на совершенствование учебно-воспитательного процесса и ориентированные на все типы ПЭВМ, используемые в сфере образования: IBM PC/XT/AT (и совместимых с ними), «Ямаха», КУВТ-86, «Электроника УК НЦ», «Корвет», «Агат» и др. Количество авторов для отдельно представляемых ПС (или их комплексов) не ограничивается. Представление материалов на конкурс — до 1 декабря 1991 г. Выпускники вузов допускаются к участию в конкурсе при условии представления материалов до 01.07.91 г.

Оплата расходов на организацию, проведение конкурса и премирование победителей будет производиться из фонда, формируемого из средств вузов-участников конкурса. Размеры и количество премий победителям будут уточнены после формирования призового фонда. Ориентировочно: две первых премии — по 1 тыс. руб., три вторых — по 700 руб., пять третьих — по 400 руб., а также несколько поощрительных премий.

К каждой конкурсной работе должны прилагаться следующие материалы:

1. Сопроводительное письмо вуза о выдвижении работ на конкурс.
2. Приложение, содержащее наименования представленных работ, список авторов с данными о них (факультет, курс, адреса и телефоны, для выпускников целесообразно указать адрес родителей).
3. Программное средство (ПС) на магнитном носителе с краткой инструкцией для пользователя в виде текста на бумаге или текстового файла.
4. Справка (в случае коллективной работы) о характере участия каждого из авторов в разработке ПС, заверенная их подписями и печатью вуза.
5. Аннотация (рекламный листок) на представленное ПС в объеме до 1 машинописной страницы с кратким изложением назначения и функциональных характеристик.
6. Тезисы доклада по теме конкурсной работы объемом 1—2 машинописных страниц (через 2 интервала) для участия в заключительной студенческой конференции.
7. Другие необходимые инструктивные и методические материалы, а также отзывы и рецензии (если имеются).

По итогам конкурса РЦ НИТО проведет для участников конкурса студенческую научно-практическую конференцию с изданием сборника тезисов (ориентировочные сроки — зимние каникулы 1992 г.).

Работы представлять непосредственно в конкурсную комиссию или направлять по адресу (с пометкой «на конкурс»):

**644099, Омск — 99, наб. Тухачевского, 14. Омский пединститут, РЦ НИТО.
Справки по телефонам: (8—381—2) 236256, 231600.**

Оргкомитет гарантирует сохранность представленных материалов, а также полностью исключает возможность несанкционированного копирования и передачу ПС в чужие руки. Право распоряжаться ПС после конкурса сохраняется исключительно за разработчиками.

В конкурсе могут принять участие студенты других вузов.

В. КИРЮХИН,
канд. техн. наук

Всесоюзные олимпиады по информатике — вчера, сегодня, завтра

Олимпиады по информатике среди школьников стали традицией. В прошлом году в Минске была проведена II Международная олимпиада, а в Харькове — III Всесоюзная. Олимпиады по информатике, в отличие от других (по таким предметам, как химия, физика, математика), находятся еще в стадии становления. Отбираются наилучшие формы проведения этих соревнований, подходы к составлению задач и оценке их решений.

Несмотря на то что олимпиады по информатике аналогичны по своей сути всем другим, тем не менее организация и проведение их с технической точки зрения значительно сложнее. Во-первых, требуется достаточно большое количество персональных компьютеров (ПК); во-вторых, необходимо соответствующее программное обеспечение; в-третьих, вся техника и программное обеспечение должны бесперебойно функционировать; и последнее, составление задач, проведение олимпиады и проверка работ участников в этом случае намного сложнее и могут быть осуществлены только квалифицированными специалистами в области информатики и вычислительной техники.

Если при проведении международных олимпиад по информатике (МОИ) вопрос об обеспечении персональными компьютерами практически не стоит, то выбор места проведения всесоюзных олимпиад начинается именно с его решения. Общее мнение — олимпиады всесоюзного и международного уровня должны проводиться на персональных компьютерах, совместимых с *IBM PC* (наиболее надежных и программно обеспеченных). Однако, как показал опыт проведения прошлых олимпиад, очень мало в стране регионов и городов, способных предоставить в распоряжение олимпиады требуемое количество таких персональных компьютеров. Если города проведения первых всесоюзных олимпиад по информатике —

Свердловск, Минск и Харьков — в той или иной степени справились с этой задачей, то организация следующих вызывает серьезную озабоченность.

Среди возможных вариантов решения этой задачи — централизованная поставка персональных компьютеров в Гособразовании СССР для проведения соревнования с последующим распределением их по учебным заведениям соответствующего региона; привлечение в качестве спонсоров ведущих наших фирм или совместных предприятий, способных на время олимпиады поставить необходимое количество ПК.

Последний путь, на мой взгляд, наиболее перспективный, но пока со стороны возможных спонсоров нет надежных предложений, хотя необходимость в этом очень большая.

Важным вопросом является форма проведения олимпиад по информатике. Сейчас олимпиада — индивидуальное соревнование. Однако возможны и другие варианты, например соревнование команд, в которых каждый член выполняет вполне определенные функции. Опыт проведения командных олимпиад уже есть. В частности, на этих принципах проводилась Международная олимпиада по программированию в Югославии в 1988 г., регулярно такого рода соревнования проводятся в Англии и других странах. Наверняка аналогичные соревнования будут проводиться и в нашей стране, хотя в организационном плане и техническом обеспечении такие олимпиады еще более дорогостоящие.

Что касается индивидуальных соревнований, то здесь возникает еще один вопрос: проводить олимпиаду в один или несколько туров, а также какова роль компьютеров в каждом туре? В нашей стране с самого начала было принято проводить олимпиады по информатике в два тура. Затем этот подход распространился и на международ-

ные олимпиады. Если I МОИ, проходившая в 1989 г. в Болгарии, была организована в один тур, то II МОИ в Минске уже состояла из двух туров. Опыт показал, что такая схема позволяет школьникам избежать случайных срывов, вызванных излишним волнением и отсутствием навыков участия в таких ответственных соревнованиях.

Много дискуссий ведется о том, как проводить оба тура олимпиады, т. е. сколько задач предлагать, какого характера, есть ли необходимость в использовании компьютеров на каждом туре. У международного жюри по этому поводу не было никаких сомнений — предлагать на туре по одной задаче, ориентированной на применение компьютера. На III Всесоюзной олимпиаде мнения жюри разошлись. Принятое большинством голосов решение о проведении двух туров с использованием ПК вызвало неоднозначную реакцию среди руководителей команд. Некоторые из них утверждали, что принятый на прошлых олимпиадах порядок проведения туров, т. е. I тур — теоретический без использования компьютеров, а II тур — практический с их использованием, позволяет, с одной стороны, уравнивать шансы на победу школьников, имеющих различный доступ к персональным компьютерам, а с другой стороны, выделить ребят, которые имеют хорошую теоретическую подготовку, но не могут быстро нажимать на клавиши.

У меня правильность принятого решения не вызывает сомнения в силу следующих причин:

в процессе решения задач по информатике этапы формализации, разработки наилучшего алгоритма и его реализации на персональном компьютере взаимосвязаны и неотделимы друг от друга, более того, разработка алгоритма не может осуществляться без учета возможностей конкретной ЭВМ;

техника работы на компьютере является неотъемлемой частью решения задачи по информатике;

проведение тура с использованием компьютеров также предполагает необходимость анализа эффективности предлагаемых школьниками алгоритмов;

уровень всесоюзных олимпиад по информатике должен соответствовать международному уровню, и если раньше проведение теоретического тура, равно как и введение в стране бескомпьютерного обучения информатике в средней школе, было связано с отсутствием необходимой вычислительной техники, то сейчас положение с персональными компьютерами существенно улучшилось, и в каждой республике есть необходимые условия для подготовки своих команд с учетом названных требований;

введение компьютерных туров несколько не ограничивает класс предлагаемых на олимпиаде задач, так как в этом случае появляется возможность самим школьникам определить, в какой мере они должны использовать компьютер для получения требуемого результата;

значительно повышается уровень объективности оценок за счет использования компьютеров при проверке решений, представленных участниками.

Наибольшей критике на олимпиадах любого уровня подвергаются предлагаемые задачи. Это и понятно. От выбранных задач зависит очень много, в конечном счете и успех самой олимпиады. Проблема эта действительно очень сложная и ответственная, однако меня всегда удивляет тот ажиотаж, который искусственно создается вокруг этого вопроса руководителями команд, особенно на всесоюзных олимпиадах. Более того, неудачное выступление участников той или иной команды тут же объясняется плохими задачами.

Конечно, задачи бывают разные. Они могут кому-то нравиться, а кому-то — нет, но нельзя забывать, что олимпиада по информатике — это не конкурс задач, а соревнование участников на лучшее решение представленной задачи. На мой взгляд, для того чтобы задачу можно было предложить на олимпиаде, она должна удовлетворять следующим требованиям:

быть не известной всем участникам; содержать оригинальную идею, требующую от участников использования нетрадиционных подходов к ее решению;

ее решение не должно базироваться только на использовании специальных знаний;

быть разумного уровня сложности.

В последнее время в значительной степени изменилось отношение и к количеству предлагаемых задач на олимпиаде. Анализируя данные, представленные в таблице, можно отметить явную тенденцию к уменьшению числа задач.

Это говорит о следующем.

Во-первых, решение задачи с использованием компьютера предполагает выполнение школьниками в полном объеме всех названных этапов. Подобрать сколько-нибудь интересные задачи, сочетающие эти этапы и удовлетворяющие перечисленным требованиям, очень сложно в силу специфики школьного курса информатики. Действительно, сложность алгоритма предполагает и сложность ее реализации на ЭВМ, при этом имеется в виду не объем программы, а процесс ее написания и отладки. Даже если такие задачи удалось составить, то отведенного

времени, т. е. четырех, а может быть, и пяти часов, оказывается явно недостаточно, чтобы школьники могли решить две и больше задач в полном объеме.

Таблица
Число задач, предлагавшихся на прошлых олимпиадах

Название олимпиады	I тур	II тур
I Всесоюзная (Свердловск, 1988)	4	2
II Всесоюзная (Минск, 1989)	3	1
III Всесоюзная (Харьков, 1990)	2	1
I Международная (Правец, НРБ, 1989)	1	—
II Международная (Минск, 1990)	1	1

Во-вторых, предлагаемые на первых олимпиадах формулировки заданий отличались четкой постановкой задачи и в основном сводились к исследованию алгоритма ее решения. При таком подходе практически исчезал этап формализации задачи и сводился к минимуму процесс программирования. Ограниченное число методов решения такого рода задач привело к тому, что авторы в этом плане в значительной степени исчерпали свои возможности. Кроме того, в этом случае олимпиада по своему характеру приближалась к математической, что вызывало немало нареканий со стороны членов жюри и участников олимпиады.

В-третьих, нельзя забывать и о культуре программирования, являющейся неотъемлемой частью информатики. Однако эти качества школьники могут продемонстрировать только в процессе решения задачи, который при этом не должен свестись только к программированию, так как олимпиада все же по информатике, а не по программированию.

Эти и другие соображения привели к тому, что у многих сформировалось вполне определенное мнение — на каждом туре предлагать не более одной задачи, как это было и сделано на II Международной олимпиаде. Однако в этом случае к перечисленным ранее требованиям к олимпиадным задачам добавляется еще одно. Условие задачи должно содержать несколько подзадач, объединенных одной идеей, но различных по уровню сложности и расположенных в порядке его возрастания, причем простейшие подзадачи (одна или две) должны быть доступны большинству участников олимпиады, в то время как наиболее сложные — только сильнейшим.

Такой подход на практике себя полностью оправдал, хотя некоторые руководи-

тели команд по этому поводу высказывают ряд опасений. В частности, они считают, что очень плохо, если ни один из участников олимпиады не выполнит всех заданий полностью. Это может даже у победителей олимпиады вызвать чувство неудовлетворенности. На самом деле, если условия олимпиады заранее доведены до сведения всех участников, то все опасения снимаются. Более того, такой подход побуждает к дальнейшему совершенствованию даже победителей.

Заслуживает внимания еще один вариант проведения олимпиад по информатике: на каждом туре предлагается не одна задача, а несколько, и каждый участник имеет право самостоятельного выбора. Однако в этом случае сразу возникают две проблемы. Первая связана с тем, как затем оценивать результаты. Практически невозможно ни до, ни после тура сказать, какая задача именно для этого состава участников сложнее или проще. Вторая проблема связана с тем, что с самого начала мы ставим участников в различные условия, нарушая принципы проведения олимпиад. Помимо этого перед участниками встает еще выбор задачи. Вместо того чтобы сосредоточиться на решении вполне определенной, они вынуждены метаться от задачи к задаче, теряя драгоценное время.

Одним из наиболее важных и сложных вопросов организации и проведения олимпиад по информатике является проверка и оценивание результатов работ участников. На мой взгляд, очень хорошо эта процедура отработана на международных олимпиадах. В частности, существует так называемая координационная комиссия, на плечи которой ложится весь процесс проверки задач. Члены комиссии строго руководствуются критериями оценки задач, рассмотренными и утвержденными международным жюри. Руководители команд, являющиеся членами жюри, также осуществляют проверку работ своих подопечных, после чего результаты согласуются с координационной комиссией. В случае возникновения каких-либо разногласий обе стороны стараются прийти к общему соглашению. Если это не удастся, спорные вопросы выносятся на рассмотрение жюри.

На всесоюзных олимпиадах эта процедура выглядит иначе. Жюри состоит из двух групп. В первую входят представители всех союзных республик и члены методической комиссии Гособразования СССР, во вторую — представители города-организатора олимпиады. В проверке задач принимают участие практически все члены жюри. После окончания проверки объявляются критерии оценки и предварительные результаты. Участникам предоставляется право апелли-

ровать к жюри. По результатам рассмотрения апелляций подводятся окончательные итоги, и только тогда объявляются победители.

Такая схема подведения итогов характерна для всех всесоюзных олимпиад. Если на олимпиадах по физике, математике, химии она вполне себя оправдала, то по информатике при такой организации возникает ряд проблем. Во-первых, очень трудно формализовать критерии оценки самих программ, а тем более алгоритмов, написанных от руки и не реализованных на ЭВМ. Во-вторых, сама проверка программ — не такое уж простое дело и требует высокой квалификации и достаточного времени у проверяющих.

Использование тестов не спасает ситуацию, так как, с одной стороны, трудно говорить об их полноте, а с другой стороны, возникает вопрос: как оценивать работу, если только часть тестов проходит либо не проходит ни один тест, когда программа содержит синтаксические ошибки или решает только часть задачи? Последний момент вызывает особые споры, так как иные руководители команд считают синтаксические ошибки незначительной погрешностью и требуют достаточно высокой оценки работы, забывая в то же время, что отладка программ является неотъемлемой частью культуры программирования и грамотное ее проведение требует времени, причем немало.

Указанные сложности приводят к тому, что при всем старании членов жюри не удается достичь абсолютного единства в оценке результатов и исключить ошибки при проверке. Механизм апелляций позволяет устранить все неточности в работе жюри, но, как показал опыт III Всесоюзной олимпиады, стремление участников во что бы то ни стало повысить себе оценку привело к тому, что члены жюри должны были выполнить за несколько часов очень большой объем работы, причем в большинстве случаев совершенно не обоснованно.

Анализ показывает, что в чистом виде перенести процедуру оценки задач, принятую на международных олимпиадах, на всесоюзный уровень не представляется возможным. Однако некоторые принципы, на мой взгляд, следует сохранить. Например, выделить из состава жюри комиссию по проверке задач, оставив за жюри утверждение критериев проверки и принятие решений по оценке результатов. В то же время необходимо предоставить руководителям команд возможность согласовывать результаты работы комиссии с результатами их собственной проверки и анализа, ликвидировав при этом механизм апелляций с возложением функций защиты интересов участников на их руководителей.

Подводя итог сказанному, можно выделить следующие основные моменты организации будущих олимпиад по информатике:

1. Олимпиада должна проводиться в два тура, причем оба тура с использованием персональных компьютеров.

2. На каждом туре участникам предлагается по одной задаче.

3. Для проведения олимпиады формируются оргкомитет, методическая комиссия, жюри и комиссия по проверке решений участников.

4. Методическая комиссия формируется приказом Гособразования СССР, и в ее состав во главе с председателем входят квалифицированные специалисты, которые делегируют республики, Москва и Ленинград. Задача этой комиссии — представить на рассмотрение жюри задачи и предложения по оценке их решений.

5. Жюри олимпиады формируется на время проведения соревнования и утверждается центральным оргкомитетом всесоюзных олимпиад. В его состав входят председатель и два заместителя, а также по одному представителю от каждой республики или региона, делегировавших команды для участия в олимпиаде. Основные задачи жюри — утверждение текстов задач, критериев их оценки, принятие решений по всем вопросам, связанным с проведением туров, и окончательное утверждение результатов олимпиады.

6. Комиссия по проверке задач формируется в основном из представителей города — организатора олимпиады с привлечением наиболее квалифицированных специалистов из союзных республик и регионов страны. Основная задача этой комиссии — объективная проверка всех работ участников и согласование результатов проверки со всеми руководителями команд. В своей работе комиссия по проверке задач руководствуется только решениями жюри олимпиады.

7. В обязательном порядке во всех командах должны быть представлены учащиеся не только выпускных классов, при этом необходимо увеличить численный состав команд и в перспективе проводить соревнования отдельно по разным возрастным группам, как это сделано на олимпиадах по математике, физике и химии.

В заключение хотелось бы несколько слов сказать о финансовой поддержке олимпиады по информатике. Опыт показал, что средств, выделяемых на проведение олимпиады централизованно, оказывается явно недостаточно. В основном их хватает только на покрытие расходов на питание, проживание и командирование представителей всех служб олимпиады. Это приводит к тому, что

большая часть работы по подготовке и проведению олимпиады проводится на энтузиазме. В результате практически невозможно привлечь к этой деятельности ведущих специалистов страны в области информатики. Особенно это сказывается на работе методической комиссии. Выход из этого положения я вижу в организации специального фонда поддержки олимпиад по информатике путем привлечения средств ведущих фирм нашей страны, совместных предприятий и кооперативов. Пользуясь случаем, хочу обратить внимание этих организаций на необ-

ходимость такой поддержки и готовность оргкомитета на сотрудничество с учетом взаимных интересов, как это делается во многих странах мира.

Многие мысли, высказанные в этой статье, неоднократно обсуждались в процессе проведения предыдущих олимпиад. Однако хотелось по этому поводу услышать и другие мнения. Мы готовы к обсуждению возникающих в процессе проведения олимпиад вопросов со всеми заинтересованными лицами и организациями и надеемся также на отклик читателей «ИНФО».

В. ФЕДОТОВ

Ивановский государственный университет

Учебные программы в стихии рынка

105

После введения ограничений в торговле столица избавилась от многочисленных гостей с котомками и тележками. Пустынно и в заповеднике застоя — ВДНХ. Словно вымер обычно шумный в школьные каникулы павильон «Народное образование». И лишь небольшое оживление в той его части, где развернулась выставка педагогических программных средств (ППС).

Увы! Это совсем не те посетители, которых ждали организаторы выставки. Все дисплеи оккупировали дети, успевшие прознать, что в разряд ППС почему-то попали и отнюдь не учебные, чисто игровые программы, точь-в-точь такие же, за которые у кооператора пришлось бы изрядно раскошелиться. Изредка отвлекают дремлющих консультантов едва ли не случайно забредшие сюда вузовские преподаватели и методисты, но практически нет учителей.

Отчего же столь стремительно иссяк интерес, который всего каких-нибудь 3—5 лет назад невозможно было удовлетворить? Безусловно, первая причина — положение на рынке программ в сочетании с нищетой системы образования. Зачем тратить время и попусту глазеть на то, что купить не по карману! А цены на компьютерные программы уже порядками отличаются от выделенных школам на их приобретение средств.

Вторая причина — отсутствие квалифицированной экспертизы программных средств. Если еще года три назад программы были оригинальны и единичны, то сейчас можно выбирать из десятка, а иногда и больше внешне одинаковых предложений. Но как предпочесть одного кота в мешке другому? И не в том беда, что одна программа считает чуть медленнее, а другая без

надобности перегружает память. Гораздо неприятнее недодуманные авторами, незаметные до поры до времени побочные эффекты, не говоря уже о вполне сознательно заложенных вирусах.

Наконец, третья причина — качество самих программ. Хотя сюжетная тематика ППС невообразимо расширилась, последние 2—3 года не дали новых идей ни в построении программ, ни в методике их использования на уроке. Коммерциализация программирования привела к тому, что усовершенствование программ шло в последние годы исключительно в направлении их защиты от несанкционированного копирования. Материалы выставки говорят об этом более чем убедительно. Положение усугубляется тем, что методически безграмотные профессиональные программисты вытеснили с рынка ППС программирующих хуже них учителей.

Интерес авторов программ можно понять прекрасно. Куда труднее понять руководителей образования, полностью пустивших этот процесс на самотек. Правда, некоторым их оправданием может служить то, что пока минпросы перестраивались в госкомобры, им было не до управления: одни безуспешно боролись за сохранение старого кресла, другие спешно подыскивали новое.

А тем временем сложилась прямо-таки парадоксальная ситуация. С одной стороны, это обилие ППС, совершенно не соответствующих запросам учителей. Программы эти тем не менее хорошо раскупаются, что объясняется некомпетентностью распоряжающихся финансами администраторов, их неустойчивостью перед обольстительными и не

всегда законопослушными коммерсантами, а также постоянным появлением компьютерных классов, неопытные хозяева которых хватают первое, что попадает на глаза. С другой стороны, практически у каждого учителя, ведущего ОИВТ хотя бы 2—4 года, есть — пусть даже очень короткие — собственные оригинальные и тщательно методически продуманные программы, которые он давно отчаялся внедрить.

Начать с того, что не ехать же для этого каждому учителю в далекий НИИ ИВТ АПН СССР. А в областных ИУУ, не говоря уже о РМК, ограничиваются только тиражированием спущенных сверху программ, да и то далеко не всегда. Но совсем в глупом положении оказывается тот учитель, который за помощью во внедрении своей программы придет в кооператив или малое предприятие. Последних немало появилось и под крышей институтов Академии педнаук, в чьи прямые обязанности входит подобная деятельность.

Сначала его программу в пух и прах раскритикуют. Того, что программа хорошо вписывается в школьный курс, удобна на уроке, достигает требуемого учебного или воспитательного эффекта, здесь оценить не способны. Зато показательно удивятся скудости видеоэффектов (нередко лишь отвлекающих внимание в ненужную сторону), отсутствию защиты от копирования (перегружающей программу без какой-либо пользы с точки зрения целей урока), съязвят по поводу Бейсика (более удобного массовому учителю, чем коды или загрузка через приобретенную специально ради 1—2 программ операционную систему). Но программу все же скопируют, а затем переработают — в меру своего разумения (см. предыдущую фразу), что, однако, даст им достаточное, с собственной точки зрения, основание для присвоения авторства. И спустя некоторое время учитель сможет втридорога приобрести бывшую свою же программу, оснащенную десятком бесполезных «академических» эффектов.

Между тем выход из создавшейся тупиковой ситуации неоднократно предлагался автором этих строк на различных семинарах и конференциях по школьной информатике еще задолго до того, как ситуация зашла в тупик (см., например: ИНФО. 1989. № 1. С. 128. П. 5 резолюции). Сейчас им воспользоваться легче, если объединить усилия начинающих вставить на ноги органов управления образованием и недавно созданных ассоциаций учителей информатики. Усугубляющаяся экономическая обстановка в стране, особенно больно ударяющая в первую очередь по социальной сфере, включая

образование, требует безотлагательных действий.

Мое предложение состоит в организации в рамках системы среднего и педагогического образования (включая университеты) общесоюзной сети свободного копирования ППС. Центрами сети должны стать кафедры и кабинеты информатики вузов и ИУУ, дома пионеров и РМК, а также школы, имеющие ЭВМ редких марок либо заметные своими успехами в информатизации. Структура связей между центрами должна учитывать как географические факторы (транспортные сообщения, административно-территориальное деление), так и функциональные интересы (по типам ЭВМ, специализацию по учебным дисциплинам).

Сеть должна обеспечивать следующий комплекс услуг: 1) возможность для автора программы беспрепятственно зарегистрировать ее в ближайшем центре сети, 2) передачу кратких аннотаций и другой информации о ППС по всей сети, 3) предоставление ППС по запросу любого абонента сети. Все это должно делаться безвозмездно, с погашением чисто технических затрат из госбюджетных средств, выделяемых на образование.

Не должны ущемляться и авторские права. Но поощрять авторов нужно не за объем программы или умение ее разрекламировать. С этой целью должен регулярно проводиться опрос пользователей о фактически используемых ими программах, их достоинствах и недоработках, а также о мотивах, по которым забракованы неиспользуемые программы. Такой опрос и есть наиболее объективная и беспристрастная экспертиза. На основании этой информации авторам лучших и наиболее используемых программ будут выплачиваться гонорары (из тех средств, которые сейчас ничтожными порциями распылены по тысячам школ). Информация же о слабых местах будет сообщаться авторам программ, чтобы они могли сделать улучшенные их версии, а также пользователям, когда речь пойдет о предостережении от неожиданных неприятностей.

Нетрудно понять, что такая сеть окажется вполне способной конкурировать с малыми предприятиями и кооперативами в сфере ППС. Прежде всего, школам выгоднее вложить свои крохотные средства именно сюда и получать за это большое количество разнообразных ППС, фактически оплачивая только лучшие из них и регулярно используемые. Выгодно это и авторам. Пострадают только те, кому прежде удавалось успешно сбывать откровенно халтурные ППС. Первоначальный же фонд ППС сети

составят те авторы, которые пока вообще остаются без вознаграждения за эту свою работу. Не вызовет сложности и разрешение конфликтов, касающихся авторства программ. Достаточно сопоставить даты их ре-

гистрации, а при необходимости — и тексты. Наконец, создание сети обеспечит возможность выбирать истинно лучшие ППС, что должно привести к качественному скачку в преподавании.

Ульяновский ПКБ АС — ваш надежный партнер, пользуйтесь его услугами

Разработка и поставка программно-технических комплексов (ПТК) и автоматизированных рабочих мест (АРМ) «под ключ», поставка прошедших испытания в условиях разных техникумов и других средних специальных учебных заведений АРМ:

*члена приемной комиссии;
сотрудника учебной части;
организации и проведения сессии;
инспектора по кадрам;
учета работы и загрузки преподавателя;
по составлению расписаний.*

Все указанные АРМы функционируют на базе ПЭВМ типа IBM PC или отечественных аналогов.

Разработка подсистем АСУ по финансовой и экономической деятельности учебных заведений.

Поставка отечественных и импортных ПЭВМ.

Доработка и адаптация программных средств.

Обучение пользователей эксплуатации поставленных АРМов.

ПКБ АС — это:

коллектив разработчиков, имеющий 20-летний опыт разработки и внедрения АС различных классов для предприятий и организаций многих отраслей народного хозяйства (промышленность, транспорт, здравоохранение, финансы, народное образование, сельское хозяйство и др.);

прочные связи с ведущими фирмами страны, работающими в области развития и применения ВТ в управлении;

богатейшая библиотека системных и функциональных программ для отечественных и импортных ЭВМ;

школа, обеспечивающая высококвалифицированную подготовку специалистов заказчика для работы в условиях функционирования АС и АРМ.

Обращайтесь к нам по адресу:

432601, Ульяновск, Главпочтамт, а/я 5165, директору ПКБ АС Федосееву А. Б.

или по телефонам: 31-95-74, 34-98-79, Лисову Б. Б. (автоматический код города Ульяновска — 842, доб. 2; следовательно, набирайте 842-2-31-95-74, 842-2-34-98-79).

всегда законопослушными коммерсантами, а также постоянным появлением компьютерных классов, неопытные хозяева которых хватают первое, что попадает на глаза. С другой стороны, практически у каждого учителя, ведущего ОИВТ хотя бы 2—4 года, есть — пусть даже очень короткие — собственные оригинальные и тщательно методически продуманные программы, которые он давно отчаялся внедрить.

Начать с того, что не ехать же для этого каждому учителю в далекий НИИ ИВТ АПН СССР. А в областных ИУУ, не говоря уже о РМК, ограничиваются только тиражированием спущенных сверху программ, да и то далеко не всегда. Но совсем в глупом положении оказывается тот учитель, который за помощью во внедрении своей программы придет в кооператив или малое предприятие. Последних немало появилось и под крышей институтов Академии педнаук, в чьи прямые обязанности входит подобная деятельность.

106 Сначала его программу в пух и прах раскритикуют. Того, что программа хорошо вписывается в школьный курс, удобна на уроке, достигает требуемого учебного или воспитательного эффекта, здесь оценить не способны. Зато показательно удивятся скудости видеоэффектов (нередко лишь отвлекающих внимание в ненужную сторону), отсутствию защиты от копирования (перегружающей программу без какой-либо пользы с точки зрения целей урока), съязвят по поводу Бейсика (более удобного массовому учителю, чем коды или загрузка через приобретенную специально ради 1—2 программ операционную систему). Но программу все же скопируют, а затем переработают — в меру своего разумения (см. предыдущую фразу), что, однако, даст им достаточное, с собственной точки зрения, основание для присвоения авторства. И спустя некоторое время учитель сможет втридорога приобрести бывшую свою же программу, оснащенную десятком бесполезных «академических» эффектов.

Между тем выход из создавшейся тупиковой ситуации неоднократно предлагался автором этих строк на различных семинарах и конференциях по школьной информатике еще задолго до того, как ситуация зашла в тупик (см., например: ИНФО. 1989. № 1. С. 128. П. 5 резолюции). Сейчас им воспользоваться легче, если объединить усилия начинающих вставать на ноги органов управления образованием и недавно созданных ассоциаций учителей информатики. Усугубляющаяся экономическая обстановка в стране, особенно больно ударяющая в первую очередь по социальной сфере, включая

образование, требует безотлагательных действий.

Мое предложение состоит в организации в рамках системы среднего и педагогического образования (включая университеты) общесоюзной сети свободного копирования ППС. Центрами сети должны стать кафедры и кабинеты информатики вузов и ИУУ, дома пионеров и РМК, а также школы, имеющие ЭВМ редких марок либо заметные своими успехами в информатизации. Структура связей между центрами должна учитывать как географические факторы (транспортные сообщения, административно-территориальное деление), так и функциональные интересы (по типам ЭВМ, специализацию по учебным дисциплинам).

Сеть должна обеспечивать следующий комплекс услуг: 1) возможность для автора программы беспрепятственно зарегистрировать ее в ближайшем центре сети, 2) передачу кратких аннотаций и другой информации о ППС по всей сети, 3) предоставление ППС по запросу любого абонента сети. Все это должно делаться безвозмездно, с погашением чисто технических затрат из госбюджетных средств, выделяемых на образование.

Не должны ущемляться и авторские права. Но поощрять авторов нужно не за объем программы или умение ее разрекламировать. С этой целью должен регулярно проводиться опрос пользователей о фактически используемых ими программах, их достоинствах и недоработках, а также о мотивах, по которым забракованы неиспользуемые программы. Такой опрос и есть наиболее объективная и беспристрастная экспертиза. На основании этой информации авторам лучших и наиболее используемых программ будут выплачиваться гонорары (из тех средств, которые сейчас ничтожными порциями распылены по тысячам школ). Информация же о слабых местах будет сообщаться авторам программ, чтобы они могли сделать улучшенные их версии, а также пользователям, когда речь пойдет о предостережении от неожиданных неприятностей.

Нетрудно понять, что такая сеть окажется вполне способной конкурировать с малыми предприятиями и кооперативами в сфере ППС. Прежде всего, школам выгоднее вложить свои крохотные средства именно сюда и получать за это большое количество разнообразных ППС, фактически оплачивая только лучшие из них и регулярно используемые. Выгодно это и авторам. Пострадают только те, кому прежде удавалось успешно сбывать откровенно халтурные ППС. Первоначальный же фонд ППС сети

составят те авторы, которые пока вообще остаются без вознаграждения за эту свою работу. Не вызовет сложности и разрешение конфликтов, касающихся авторства программ. Достаточно сопоставить даты их ре-

гистрации, а при необходимости — и тексты. Наконец, создание сети обеспечит возможность выбирать истинно лучшие ППС, что должно привести к качественному скачку в преподавании.

Ульяновский ПКБ АС — ваш надежный партнер, пользуйтесь его услугами

Разработка и поставка программно-технических комплексов (ПТК) и автоматизированных рабочих мест (АРМ) «под ключ», поставка прошедших испытания в условиях разных техникумов и других средних специальных учебных заведений АРМ:

*члена приемной комиссии;
сотрудника учебной части;
организации и проведения сессии;
инспектора по кадрам;
учета работы и загрузки преподавателя;
по составлению расписаний.*

Все указанные АРМы функционируют на базе ПЭВМ типа IBM PC или отечественных аналогов.

Разработка подсистем АСУ по финансовой и экономической деятельности учебных заведений.

Поставка отечественных и импортных ПЭВМ.

Доработка и адаптация программных средств.

Обучение пользователей эксплуатации поставленных АРМов.

ПКБ АС — это:

коллектив разработчиков, имеющий 20-летний опыт разработки и внедрения АС различных классов для предприятий и организаций многих отраслей народного хозяйства (промышленность, транспорт, здравоохранение, финансы, народное образование, сельское хозяйство и др.);

прочные связи с ведущими фирмами страны, работающими в области развития и применения ВТ в управлении;

богатейшая библиотека системных и функциональных программ для отечественных и импортных ЭВМ;

школа, обеспечивающая высококвалифицированную подготовку специалистов заказчика для работы в условиях функционирования АС и АРМ.

Обращайтесь к нам по адресу:

432601, Ульяновск, Главпочтамт, а/я 5165, директору ПКБ АС Федосееву А. Б.

или по телефонам: 31-95-74, 34-98-79, Лисову Б. Б. (автоматический код города Ульяновска — 842, доб. 2; следовательно, набирайте 842-2-31-95-74, 842-2-34-98-79).

Представьте себе,

что у вас есть лазерный принтер и VGA-монитор. Следующая задача более легкая: представьте, что на этом мониторе вы сделали некий чертеж, который нужно вывести на принтер. Учтите, что VGA — это множество цветов, но только 640×480 точек, а лазерный принтер — только один цвет, но зато 12 точек на миллиметр, около 2400×3600 точек на обычном листе бумаги...

Прекрасная идея пришла в голову Ю. Плугатыреву: в битах цвета кодировать дополнительную информацию, позволяющую одну VGA-точку делить на четыре для принтера. В подробности вдаваться, видимо, не стоит — ведь у вас нет такой аппаратуры? — но... В биты цвета можно записывать и другую информацию. В частности, текстовую. Иными словами, черно-белый рисунок на экране монитора (не важно какого, даже черно-белого — лишь бы графический редактор подразумевал возможность использования цвета) может содержать в себе текст, нисколько не увеличивая при этом размер своего образа в ОЗУ или на внешнем носителе.

А как эту идею реализовать практически — придумайте сами.

Библиотека будущего

Используя новейшие технологии, дающие неограниченный доступ к знаниям со всего мира, новая Национальная библиотека Франции станет первой библиотекой XXI в.

Объявляя в 1988 г. свой план, Президент Франции Миттеран сказал, что открывающаяся в 1995 г. библиотека должна «охватить все отрасли знания, быть доступной для каждого из любого удаленного пункта, использовать наиболее современную технологию и иметь связь с другими крупными библиотеками Европы».

Концепция библиотеки на основе гигантских возможностей современных компьютеров и телекоммуникационных сетей, в которой полностью переработана сама идея библиотеки как «хранилища книг и журналов», потрясает. Новая библиотека состоит из четырех прозрачных стометровых 20-этажных башен, каждая из которых будет хранить по 7 млн. документов и тысячи видеопленок, кассет и компакт-дисков.

ЧТО МОЖЕТ ЭВМ

Благодаря техническому прогрессу окажется возможным обращение не только к печатным изданиям и электронным базам данных, но и к фильмам, телепередачам и музыкальным записям.

Научные работы, статьи из периодики и книги будут поступать на мощные главные компьютеры библиотеки. Соединяясь по сети из любого места Европы с одним из 300 ее терминальных компьютеров, читатель (впрочем, это будет уже не обязательно читатель) сможет быстро обращаться к гигантскому количеству информации.

Очень важно, что этим обеспечивается сохранность любых документов. Вместо изучения документа буквально вручную люди смогут увидеть и обработать оригиналы исторических работ, получая их на свой домашний компьютер через высокоскоростную цифровую телекоммуникационную систему связи «МИНИТЕЛ», которая уже теперь доступна по всей Франции. Создание электронной библиотеки имеет и много других преимуществ, не последнее из которых — использование способности ЭВМ сортировать и классифицировать данные. Так, традиционный библиотечный каталог классифицирует книги только по фамилии автора, заглавию и тематическому содержанию. Компьютерный каталог позволит получить любые варианты классификации, начиная от даты публикации до места издания или длины документа. «Читатель» сможет иметь любого типа ссылки на те или иные источники, производить поиск информации среди полного текста миллионов книг и заметок по любому вопросу, будь то Французская революция или первое испытание атомной бомбы.

Создатели этого очередного чуда света надеются, что к 1995 г. около 300 000 книг, что составляет 3% фондов действующей

Национальной библиотеки, будут введены в компьютеры для начала опытной эксплуатации.

А когда строительство будет закончено, эта библиотека будет служить символом Франции наравне с Эйфелевой башней.

Часы, приемник, компьютер

Все эти и по отдельности полезные предметы, объединенные вместе, превращаются в персональную систему связи, позволяющую не только быстро связаться с нужным абонентом, но и передать ему короткое сообщение или номер телефона, по которому необходимо позвонить.

Новый прибор, который получил название «часы для сообщений» Message Watch, — результат работы двух фирм: Hattorri Seiko, которая спроектировала новый комбайн, и AT&E, разработавшей его программное обеспечение, тексты сообщений и систему их кодирования.

Наручный прибор и впрямь мало отличается от обычных электронных часов. На его привычном жидкокристаллическом дисплее можно видеть дату и время дня любой часовой зоны. Но, кроме того, «часы» могут с помощью встроенного миниатюрного приемника принимать сообщения, хранить их неограниченное время в собственной памяти и демонстрировать по первому требованию на экране дисплея.

Размеры сообщения невелики: всего две строки по восемь символов в каждой. Но этого достаточно, чтобы попросить позвонить домой, на работу или по указанному телефону, напомнить о встрече или сообщить о начале важной операции. Поступающее сообщение начинается с гудка, который, впрочем, может быть отключен. Всего «часы» могут принять и хранить до восьми сообщений. Каждое устройство имеет индивидуальный код, не позволяющий принимать «чужую» информацию.

Чтобы послать сообщение владельцу таких часов, необходимо по обычному телефону позвонить в специальный центр, где за определенную плату закодируют вашу просьбу и передадут в эфир. И где бы ваш абонент ни был, он обязательно вас услышит.

Е. ПОЛАТ

докт. пед. наук,

А. ЛИТВИНОВА

Информационные технологии в зарубежной школе

Определение основных направлений современной школьной политики, претворение их в жизнь, а также прогнозирование развития школьного образования предполагают тщательный анализ зарубежного опыта развитых стран, достигших в этой области определенных успехов.

В настоящее время актуальность такого анализа ощущается в нашей стране особенно остро, в связи с явным отставанием в решении проблем, поставленных обществом перед школой и педагогикой. Это в полной мере можно отнести и к сфере развития учебно-материальной базы (УМБ) общеобразовательной школы.

Наметившийся в 70-е гг. глубокий разрыв в уровнях развития УМБ нашей школы и школ западных стран в последние годы стал приобретать очертания пропасти: в учебных заведениях США, Великобритании, ФРГ, Канады, Франции, Австралии, Японии, Швеции и других государств начали использоваться такие средства новых информационных технологий (НИТ), как персональные компьютеры, интерактивное видео (в том числе основанное на использовании лазерных оптических видеодисков), телеконференции, электронная почта, а в последнее время и системы искусственного интеллекта, технология *CD-ROM*. Ничего из перечисленных средств, кроме компьютеров, в наших учебных заведениях (и тем более школах) пока нет.

Школьная практика стран Запада располагает достаточным опытом применения различных средств НИТ, на наш взгляд, еще и потому, что там отсутствует монополия одного ведомства на разработку, производство и внедрение того или иного средства обучения. В этой области, как и в других, господствует рынок со своими законами, предоставляющий учителю возможность широкого выбора необходимых ему и качественных дидактических средств из их многочисленного перечня. Этот аспект нам также

нельзя упускать из виду при изучении зарубежного опыта, учитывая, что дальнейшее развитие УМБ нашей школы будет во многом определяться новыми условиями нарождающейся рыночной экономики.

Особый интерес представляет оценка эффективности использования отдельных видов средств обучения, которая дана зарубежными специалистами в их последних работах.

Давая сравнительную характеристику отдельным видам средств обучения [1] с точки зрения их эффективности, многие зарубежные специалисты отмечают довольно низкий рейтинг учебного телевидения в т. ч. и кабельного. «Попытки учебного телевидения заменить классно-урочное обучение не увенчались успехом», — пишут по этому поводу американские исследователи (Werkman, 1976; Shorestein, 1978). Хотя следует признать, что ряд университетов в США успешно использовали программы коммерческого телевидения для заочного обучения. Открытый университет в Великобритании на протяжении более 10 лет ведет исследования по организации заочного обучения студентов с помощью телевидения, подготовив за это время в целом свыше 1500 телепрограмм (по 35 ч в неделю). И все же большинство специалистов склонны рассматривать опыт учебного телевидения как неудачу. То же самое можно отнести и к учебному кино.

Результаты эксперимента, проведенного американскими учеными по определению сравнительной эффективности учебного телевидения, видеозаписи, интерактивного видео и видеотекста на базе курсов подготовки специалистов корпорации General Motors, свидетельствуют о том, что наименее эффективным из перечисленных средств оказалось трансляционное учебное телевидение.

В начале 80-х гг. в США, Великобритании и некоторых других странах была апробирована информационная система видео-

текса [2]. Некоторые фирмы, разработчики подобных программных средств, предложили образовательную систему, включающую, наряду с обучающими программами, чисто информационные материалы: энциклопедии, библиографические справочники и пр. Однако попытки включить эту технологию в систему образования не увенчались успехом, по крайней мере в США. Но они продолжают успешно функционировать в качестве инструмента для поиска нужной информации (библиографические справочники и пр.) или в качестве доски объявлений (в системе электронной почты).

К средствам обучения, подтвердившим свою эффективность в течение ряда лет, можно отнести *видеозапись*. Начиная с 1975 г. в системах образования США, а также стран Западной Европы с неизменным успехом используется видеозапись самых различных жанров. Она имеет самый высокий рейтинг среди средств обучения. В США стоимость ежегодно выпускаемой сегодня видеозаписи составляет примерно 4 млн. долларов.

Такому успеху способствовали доступность видеозаписи для широкого круга учащихся, простота использования, занимательность и, конечно, качество самих программ. Видеозапись дает возможность выбора программы, времени ее просмотра, а также определения той или иной дидактической функции конкретной видеозаписи в учебном процессе.

Вместе с тем есть целый ряд новейших средств обучения, эффективность использования которых в системах образования, несмотря на достаточно обнадеживающие результаты, подлежит дальнейшему тщательному исследованию для получения обоснованных выводов педагогической целесообразности их включения в состав УМБ современной общеобразовательной школы (или другого типа учебного заведения). К таким средствам обучения относятся прежде всего *персональные компьютеры, интерактивное видео, телекоммуникации, технология CD-ROM, программы искусственного интеллекта*.

Так, телекоммуникации, в частности телеконференции, в связи со значительным снижением стоимости спутниковой связи и всего необходимого оборудования стали доступны большому числу организаций, в том числе и учебных заведений. Правда, телеконференции пока еще не нашли такого широкого распространения, как это предсказывали специалисты. Но во многих случаях, в том числе и в системах образования, когда необходимы срочные контакты сразу с несколькими участниками работы для обсуждения какого-то вопроса, принятия срочного согласованного решения, проверки и ис-

полнения, инструктажа и пр., телеконференции оказываются просто незаменимыми. Особенно ощутима их эффективность для международных контактов. В этом случае их функции выходят далеко за рамки просто деловых контактов. Здесь уже огромное значение приобретают социальный, культуроведческий и даже политический факторы.

Технология телекоммуникаций, как и любого вида информационной технологии, недешева. И все же большое количество школ в странах Запада начинают использовать телекоммуникации в своей практике. В США свыше 6000 городских школ и 26 000 районных оснащены модемной связью. 35 штатов включены в телекоммуникационные программы, многие из которых базируются на спутниковой связи. В первую очередь это касается отдаленных районов. Научные исследовательские центры разрабатывают специальные телекоммуникационные проекты для телеконференций, электронной почты. Так, исследовательский центр Университета штата Огайо разработал экспериментальный проект телеконференций специально для школ. Этот проект был апробирован в школах штатов Арканзас, Пенсильвания и Флорида. Руководит экспериментальным проектом Консорциум по разработке дидактических материалов для профессионального и заочного обучения. Консорциум был организован в 1979 г. с целью подготовки методических материалов для повышения квалификации руководящих работников специального технического и заочного образования, а также для апробирования новых технологий обучения. Используется асинхронная система, позволяющая партнерам вносить свою информацию на доску объявлений и соответственно считывать поступившую информацию в любое, удобное для них время. Учащиеся могут получить необходимую консультацию от преподавателя. Преподаватель (или консультант, в качестве которого может выступать профессор университета, научного центра), со своей стороны, может контролировать работу учащихся, давать советы и т. д. Большую эффективность дают телеконференции и при повышении квалификации работников образования. Так, 14 администраторов из 15, прошедших курс переподготовки с помощью телеконференций, заявили, что они будут рекомендовать своим коллегам воспользоваться этой технологией. 11 из 15 участников эксперимента заявили, что только телеконференции позволили им принять участие в курсах переподготовки.

В экспериментальном проекте электронной почты и телеконференций, проводив-

шихся в университете штата Вирджиния (1987—1990), участвовало 80 % профессоров университета, более 50 % студентов, 60 % учителей и администраторов школ региона. Аналогичные эксперименты проводились в университетах Симон Фрейзер в г. Ванкувер (Британская Колумбия, 1988), в Канаде (Канадский проект Exchange), включивший 3200 участников, 1300 из которых стали активными пользователями сети. Все эти эксперименты показали преимущества телекоммуникационных технологий по сравнению с обычным телефоном и почтой, заключающиеся в большем удобстве для пользования (больше возможностей), меньшей стоимости за одинаковый объем информации.

Наибольшее преимущество телекоммуникации имеют в тех случаях, когда аппаратура установлена дома или в классе, а общение осуществляется непосредственно, т. е. синхронное.

В Великобритании все университеты и научные центры страны объединены в единую телекоммуникационную сеть Janet, что создает наиболее благоприятные условия для учебной и научной работы специалистов.

Вполне обыденным и привычным средством обучения в начальной, средней школе, профессиональном образовании многих зарубежных стран стали *персональные компьютеры*. Поток статей, монографий, описывающих разнообразие подходов к использованию компьютеров в обучении, огромен. Например, по данным австралийского алфавитного каталога за 1988 г., количество статей, посвященных компьютерному обучению в Австралии, в 8 раз превышало количество статей по другим проблемам педагогической технологии.

Несмотря на очевидную популярность персональных компьютеров в системах образования разных стран мира, нерешенных проблем, связанных с компьютерной технологией, еще много. И главные из них, пожалуй, следует отнести к программному обеспечению, с одной стороны, и к эргономической проблеме «человек — машина», с другой.

Большинство исследователей считают необходимым продолжить научный поиск наиболее эффективных способов использования компьютера в практике преподавания. До сих пор большая часть обучающихся программ — это, по сути, модификация программированного обучения 60-х гг. (линейного или разветвленного программирования), что чрезвычайно обедняет возможности этого уникального средства обучения. Очень мало программ моделирования, программ искусственного интеллекта.

Отсюда некоторое разочарование части учеников, родителей, учителей в отношении возможностей компьютера для интеллектуального развития молодежи. Результаты опросов, проведенных среди французского населения в последние годы с целью уточнения их отношения к информатике, обнаруживают наличие самых полярных мнений.

По данным 1985 г.:

энтузиасты компьютеризации составляли — 29 % опрошенных;

проявляющие интерес — 35 %;

недовольные — 11 %;

враждебно настроенные — 10 %.

Правда, значительное влияние на результаты оказал возраст интервьюированных. Так, среди молодых людей в возрасте от 15 до 19 лет энтузиастов оказалось около 42 %, в группе людей 50-летнего возраста — не более 20 %.

Не случайно основными проблемами Нового национального плана по информатике во Франции (1987) стали:

создание высококачественного в дидактическом плане программного обеспечения и поиск возможных путей для решения поставленных в этой области задач;

подготовка преподавателей, позволяющая педагогам самим участвовать в составлении (или модификации) программного обеспечения и более эффективно использовать его в учебном процессе.

Аналогичные задачи стоят практически перед всеми исследователями в этой области в США, ФРГ, Японии, Канаде, Австралии, Южной Корее.

Значение компьютера для системы образования еще не полностью осознано педагогами. Нужны дальнейшие исследования. Нельзя, однако, не согласиться с мнением канадских исследователей М. Леклерк, Л. Дюбюк и И. Бегином о том, что если персональный компьютер станет решением проблемы для очень небольшого дополнительного количества школьников (скажем, для 20 %), то уже ради этого стоит им заинтересоваться всерьез [3]. Однако для этого, как указывают те же исследователи, недостаточно ввести компьютеры и программное обеспечение в школы. Необходимо, чтобы учителя прониклись идеей их использования. Компьютеры и программы должны утратить свою искусственность, если мы хотим, чтобы они стали неотъемлемой частью процесса обучения. Кроме того, необходима интеграция информационной технологии в арсенале педагогического инструментария, возрождение идей индивидуализированного обучения.

Следует заметить, что наметились вполне четкие тенденции в определении концеп-

туальных направлений компьютеризации образования.

Курс программирования занимает все более скромное место в этой концепции, уступая место подготовке пользователей ПЭВМ, умеющих работать с текстовыми редакторами, базами данных (и создавать их), электронными таблицами, телекоммуникацией. В последнее время актуальным становится овладение навыками пользования технологией гипертекста (hypertext) и гипермедиа (hypermedia).

Система гипертекста позволяет пользователю или группе пользователей соединять разрозненную информацию, создавать аннотации текстов, постраничные или другого рода ссылки, отсылающие читателя к библиографии или источникам информации, и пр. Соединительные связки (links) позволяют читателю быстро просматривать документ за документом.

12 Система гипермедиа является, по сути, расширением возможностей гипертекста. С ее помощью пользователь может создавать целостные блоки материалов, содержащие текст, статичную, динамичную графику, видео, звук, музыку и т. д.

Наиболее впечатляющим по своим дидактическим свойствам средством обучения, появившимся в последнее десятилетие, служит *интерактивное видео*, представляющее собой соединение возможностей компьютера и видеодиска, хотя его распространение в школах Запада не столь стремительно, как это предсказывал, например, Айкенберг в 1987 г. За 1986—1987 гг. всего 14,7 % американских учреждений (не только в системе образования) использовали эту технологию, причем очень немногие из них начали систематически применять ее в качестве информационного средства. Препятствием к широкому распространению интерактивного видео является высокая стоимость программного обеспечения и особенно оборудования, отсутствие стандартизированных средств. В настоящее время практически только начинается широкое освоение этой технологии, во-первых, в связи с попытками разрабатывать собственное программное обеспечение; во-вторых, в связи с некоторым понижением цен на видеодиски и аппаратуру. Дидактические возможности интерактивного видео просто уникальны. С помощью этой технологии возможны любые сочетания звука, изображения (текстового, иллюстративного), моделирования, работа в диалоговом режиме, различные манипуляции с графикой, текстом, сочетание иллюстративных и графических способов изображений, причем пользователь может сам активно

вмешиваться в видеоряд.

Однако, как справедливо отмечает американский методист Ричард Элан Смит [4], любые средства обучения, включая и НИТ, являются лишь средством реализации методической идеи. В этом отношении видеодиск как средство обучения не представляет исключения. Однако каждое средство обучения обладает определенными дидактическими свойствами, которые в соответствии с намеченной учебно-воспитательной задачей определяют его дидактические функции. Дидактические свойства лазерного видеодиска значительно шире, чем, скажем, учебного фильма, видеоленты, слайдов и звуковых пособий. Соединение с компьютером позволяет еще в большей степени расширить его дидактические возможности, прежде всего за счет индивидуализации интерактивной методики. Разумеется, следует иметь в виду, что как автономное использование видеодисков, так и их сочетание с компьютером должны органично вписываться в используемую в практике обучения методическую концепцию.

Пока, как уже указывалось выше, системы интерактивного видео достаточно дороги. Одна такая система включает цену компьютера, программного обеспечения, видеопроигрывателя, видеодисков. Обычная цена одной такой системы (станции) для одного рабочего места колеблется в пределах от 3300 до 8000 долларов.

Для школьного образования (начального и среднего) в настоящее время выпускаются две категории видеодисков:

1. Используемые автономно для групповой работы. В этом случае видеодиск сочетает в себе свойства кинофильма, видеозаписи, слайдов и звукозаписи. Компьютер может подключаться эпизодически, чтобы автоматизировать данный способ применения видеодиска.

2. Используемые в комплексе с компьютерной программой управления диском, которая «записана» на этот же диск, специальные видеоплееры, соединенные с компьютером в единую систему, считывают эти программы, позволяя индивидуализировать интерактивную методику.

Программное обеспечение для интерактивного видео, как правило, охватывает либо весь курс учебного предмета, либо широкий спектр проблем той или иной науки. Таковы, например, интерактивные видеодиски по физике «Physical Science Curriculum», по биологии «Principles of Biology», по астрономии и освоению космического пространства «The Earth Science», по математике «Mastering Rations» и др. Имеются и чисто информационные видео-

диски, например, один такой видеодиск включает 20 томов академической американской энциклопедии «Электронная энциклопедия» (свыше 30 000 статей, 110М байт информации по различным областям науки, техники, технологии).

Поиск любой информации осуществляется мгновенно благодаря специальному устройству (The Knowledge Retrieval System). Любая информация может быть тут же распечатана на принтере. Думается, подобные системы будут доступны сначала преимущественно медиатекам, обслуживающим несколько школ, прежде чем они попадут на конкретный урок.

Первые результаты исследований эффективности интерактивного видео в системе народного образования (об одном из них упоминалось в данном анализе) по сравнению с другими видами видео (телевидение, видеозаписи, телетекс) показывают, что данное средство информационной технологии в силу уникальности своих дидактических свойств имеет большое будущее, если его стоимость снизится до уровня, доступного системам образования. Поэтому и важно продолжить педагогические исследования в этом направлении.

В самые последние годы среди средств информационной технологии появились *оптические компакт-диски*, на которые записываются не только звуковая информация, но и компьютерные обучающие программы, видеоинформация, причем в динамике, графические изображения. Новое средство получило название CD-ROM (Compact disk — READ ONLY MEMORY). Несмотря на то что первые звуковые компакт-диски появились в 1983 г., они захватили 34 % звукозаписи в США. CD-ROM технология используется для создания больших баз данных. Один оптический диск содержит 550 млн. знаков, или 250 000 страниц машинописного текста, что в тысячу раз превышает объем обычного гибкого диска.

Технология CD-ROM позволяет создавать целые библиотеки информации, обеспечивая при этом великолепную графику, динамичное видео, звук, текстовую информацию, что не под силу ни одному из компьютеров, используемых в настоящее время в школах.

Дидактические свойства оптического диска настолько уникальны, что перспективы его внедрения в систему образования не вызывают сомнения. Речь идет о тщательном исследовании дидактических функций новой технологии, определения ее места в учебном процессе и более широко — в системе непрерывного образования. Индексация, позволяющая идентифицировать рас-

положение каждого слова, цифры, любого знака в базе данных, дает возможность пользователю производить мгновенный поиск нужной информации, а также делать необходимые вставки в виде отдельных слов, цифр, абзацев и пр. В соединении с быстрым действием текстового редактора микрокомпьютера обеспечивается уникальное разнообразие дидактических возможностей использования данной технологии для целей образования. Первые попытки были связаны со школьными библиотеками и медиатеками*. Школьные библиотеки уже достаточно широко используют данную технологию в своей практике, и прежде всего для создания баз данных, различного рода каталогов. Поиск нужной литературы по такому каталогу осуществляется практически мгновенно. Школьные библиотеки, медиатеки оборудуются специальными рабочими местами для работы с CD-ROM диском. В распоряжение школ поступают диски, содержащие, например, академическую американскую «Электронную энциклопедию», Энциклопедию естественных наук и технологий, Словарь научных и технических терминов и т. д.

Встает вопрос о способах сочетания технологии CD-ROM с обычными обучающими компьютерными программами. В настоящее время ведутся интенсивные поиски наиболее рациональных способов такого сочетания. Следует иметь в виду, что CD-ROM требует специального программного обеспечения поиска информации в отличие от обычных звукового или видеооптического дисков, используемых без него. Почти все CD-ROM диски имеют собственную программу поиска информации, записанную либо на флоппи-диске, либо прямо на этом же компакт-диске. Американский специалист Ричард Дж. О'Коннор [5] предложил решить эту проблему, обеспечив вход в программу CD-ROM прямо из модуля обучающей программы (CAI). Таким образом, пользователь имеет возможность, работая с обучающей программой, в случае необходимости входить в базу данных CD-ROM. По окончании работы с ней учащийся может мгновенно вернуться в обучающую программу на то место, где остановился.

CD-ROM технология используется и как автономная обучающая система. В этом

* Под медиатекой понимается информационный центр, содержащий фонды печатной продукции (библиотека), видео и звуковой (диск, фоно), компьютерные программы, а также обеспечивающий благоприятные условия для пользования и обработки данной информации.

случае собственно компьютерная программа (CAI) используется в качестве как бы посредника между учащимися и CD-ROM.

Соединение собственно компьютерных программ и технологии CD-ROM открывает новую эру в компьютерном обучении. В настоящее время особую актуальность приобретают исследования в данной области.

4
Весьма перспективными для школьного образования представляются системы CDI (Compact disk interactive), DVI (Digital Video interactive), CDV (Compact disk video). Разница между этими технологиями заключается в способе кодирования информации, доступа к ней и воспроизведения на экранах мониторов или дисплеев. CDI содержат для обычного домашнего телевизора примерно на 5 мин демонстрации видеоизображений и на 20 мин звучания музыкального материала. Специальных программ для образования на CDI пока еще не существует. CDI содержит лишь статичные изображения, но количество таких изображений разных видов может быть достаточно значительным — тысячи изображений в сочетании со звуком и различным текстовым материалом. Эта технология доступна как для домашнего, так и для школьного пользования (уже есть учебные программы на CDI). Надо сказать, что в настоящее время примерно 33 % учащихся общеобразовательных школ в Великобритании имеют домашние компьютеры, и, соответственно, для них подобная технология в скором времени окажется вполне доступной не только в школе, но и дома. Что касается технологии DVI, то это полностью цифровая система, но она имеет возможности, схожие с аналоговыми системами лазерных видеодисков, о которых речь шла выше. Эта технология совместима с компьютерами IBM AT. Такой диск содержит информацию (в том числе и динамичную) на 70 мин демонстрации на экране компьютера, включает текст, звук, изображение. Вся информация на подобных дисках может редактироваться.

Актуальными являются и исследования в области *искусственного интеллекта*, начатые еще в 1980 г. Отдельные программы искусственного интеллекта достаточно широко используются в различных странах мира, но скорее в экспериментальном плане, чем в обычной, реальной практике.

Необходимы исследования и в области

психологии коммуникаций в условиях широкого использования средств информационных технологий: в каких случаях учащиеся предпочитают общение с живым собеседником любым видам информационной технологии; какое влияние оказывает информационная технология на интеллектуальное и творческое развитие учащихся, а также на коммуникативные способности детей, на гуманизацию и гуманитаризацию образования в целом? Эти проблемы имеют непосредственный выход практически на все аспекты воспитания и образования молодежи.

Вместе с тем важно понять, каким образом мы можем решать проблему совершенствования учебно-материальной базы современной школы, всей системы непрерывного образования. Проблема чрезвычайно сложная. Найти выход из наблюдающегося в стране кризиса народного образования, по всей видимости, удастся лишь на пути скорейшего, но, по возможности, экономного, перехода на современную учебно-материальную базу. Очень важно тщательно взвесить, проанализировать относительно каждого учебного предмета, какие конкретно из ныне используемых СО могут оказаться полезными и завтра, какие требуют модернизации, а от каких можно безболезненно отказаться и направить освободившиеся материальные вложения на развитие новых информационных технологий.

Л и т е р а т у р а

1. *Полат Е. С.* Новейшие средства информационной технологии в системе образования ведущих капиталистических стран // Проблемы использования современных технических средств обучения в школьном образовании в СССР и за рубежом. — М., 1990.
2. *Полат Е. С.* Проблемы использования компьютера в системе образования развитых капиталистических стран // Информатика и образование. 1987. № 4.
3. *Леклерк М., Дюбук Л., Бегин И.* Оценка педагогического программного обеспечения в Канаде // Prospekts. 1987. № 4.
4. *Smith R. A.* Videodisk — The Next Temptation // The Computing Teacher. 1990. Febr.
5. *O'Connor R. J.* CD-ROM Technology: A New Teaching Tool // The Computing Teacher. 1989. nov.

Компьютеры в обучении: учит ли история?

Блестящая статья А. Борка, опубликованная в «ИНФО» № 5 за 1990 г., несомненно, произвела большое впечатление на читателей, в том числе и на меня. Состояние дел в компьютеризации нашего образования не поддается не только критике, но и какой-либо приемлемой классификации. Диапазон используемой техники поражает воображение — от полного ее отсутствия и самодельных компьютеров до современных моделей зарубежных фирм с выходом на спутниковую связь. Раздумья на тему программного обеспечения вызывают только пессимизм.

Отовсюду слышатся возгласы: «Дайте, дайте, дайте... нам!!!» Не всем, но некоторым дают, и ... все возвращается на круги своя. Школа действительно не знает, что делать с компьютерами. Не буду останавливаться на доводах и аргументации учителей — они прекрасно изложены А. Борком, и все, что им описано, имеется в наличии с различными вариациями.

На мой взгляд, существует идеологическая проблема, незамедлительное решение которой архиважно. Современный учебный процесс призван решать большое количество разнообразных задач. Учитель в силу своего таланта, опыта и других качеств ищет пути к их решению. Очевидно, что при этом возникает много проблем, трудностей и т. д. В вычислительной технике большинство учителей видит средство их преодоления, т. е. инструмент, который поможет повысить качество обучения, решить проблему индивидуализации и т. д. Но! В любом случае внедрение компьютеров в учебный процесс (и, как следствие, их использование) мыслится для поддержки стандартных учебных курсов. Вот здесь, на мой взгляд, и зарыт корень зла. В Академии Платона основу составляли парная и индивидуальная формы обучения. А. Я. Коменский ввел в учебный процесс групповую. И в том, и в другом случаях (если отвлечься от исторических и социально-экономических условий) происходил кардинальный пересмотр содержания учебных дисциплин. Перестройка учебного материала носила революцион-

ный, в отличие от эволюционного развития в рамках определенной организационной формы, характер. Новая форма, новые технологии требовали иного содержания.

Мы находимся на пороге нового этапа в развитии процесса обучения. Использование компьютерной техники (со всеми ее достоинствами и недостатками) в учебном процессе требует в первую очередь перестройки мышления учителей, содержания учебных дисциплин и, как следствие, преобразования всего учебного процесса. Не удержусь от цитирования А. Борка: «...единственный путь эффективного использования компьютера состоит в разработке совершенно новых курсов, с самого начала ориентированных на применение всех учебных сред, включая новейшие, интерактивной технологии». И эти курсы должны быть интегрированными в лучшем смысле этого слова. Попробуйте определить, чего больше (математики, физики или интеллектуальной и творческой работы ученика) в следующих задачах: «Оцените время, за которое замерзнет пруд» или «С какой скоростью должен лететь теннисный мяч, чтобы разбить стекло?». Компьютер здесь становится не просто средством, а и соучастником процесса обучения, его полноправным членом наряду с учителем и учеником. Понятно, что такая перестройка идеологии учебного процесса потребует больших усилий и затрат. Но на них надо пойти, иначе мы смиренно отработаем все этапы, описанные Борком, и уткнемся в ту же стену, что и другие страны, только значительно позже их. А тенденции к этому имеются.

Во-первых, распределение техники. Кому поступает новая техника (в частности по проекту PILOT SCHOOL)? Довольно очевидно, что не самым интересным и творчески работающим учителям, хотя бывают и исключения. Значит новая техника будет просто эксплуатироваться, со всеми вытекающими отсюда последствиями (см. цитированную статью), а не стимулировать перестройку учебного процесса. Во-вторых,

программное обеспечение. В последнее время появилось огромное количество различных педагогических программных средств, направленных на компьютерную поддержку учебного процесса. Но все они, как близнецы братья. В их основе лежит древовидная схема организации связи между блоками программы (принцип «ДА—НЕТ»), что в конечном случае может помочь выработке только алгоритмического мышления. Отличаются же они только насыщенностью цветом и подсказок, красочностью оформления и эффективностью подачи. Но это еще половина беды. Хуже то, что разрабатываемые средства, в частности программные оболочки и среды, предназначены для заполнения текстовой и графической информацией с последующим ее просмотром и каким-либо реагированием (в большинстве случаев по типу: вопрос — ответ). Но тогда нужно признать, что главное в этих средствах — контроль, и не говорить красивых слов об интеллектуальном и эмоциональном воздействии на ученика, поскольку они довольно быстро затухают. Оболочек же, дающих возможность решать поисковые или исследовательские задачи (например: «Аналитическое построение графика функции», «Изучение движения тел различной формы в жидкости» и т. д.) автору, к сожалению, встретить не удалось. И уже совсем плохо то, что учителям очень часто навязывают кота в мешке. В частности, распространяемые через учебный коллектор пакеты программ по физике — откровеннейшая подделка, могущая вызвать лишь отвращение

к предмету, к тому же безобразно оформленная. Куда смотрела экспертиза? Кому это выгодно? Вообще экспертиза педагогических программных средств самое больное место, и требует незамедлительного рассмотрения и принятия жестких решений, ставящих заслон всяческой халтуре, буквально «прущей» на школу.

В-третьих, подготовка учителей. Очевидно, что просто учителю физики или математики (аналогично других предметов) осуществлять компьютеризацию учебного процесса очень тяжело. Нужна основательная переподготовка (с перестройкой идеологии образования!). В Марокко обучение учителей информатики из математиков и физиков занимает 2 (два!) года с сохранением всех благ и отрывом от производства. А у нас? Одни слезы.

И тем не менее. Несмотря на все сложности, у нас много талантливейших учителей, появляются красивые идеи и задумки. Но только объединив усилия, можно решить архисложную задачу компьютеризации учебного процесса. Очень важна здесь координирующая роль Ассоциации учителей математики, физики, информатики и других предметов.

И в заключение еще одна цитата из А. Борка: «Система образования испытывает огромные трудности, и справиться с ними можно лишь заново пересматривая учебные курсы и школьное обучение в целом».

С. ФЕОФАНОВ,
Ленинград

О физической неточности

В алгоритме и программе, предложенных Э. Гальвасом и В. Фастом в статье «Физические задачи в курсе информатики» (ИНФО. 1990. № 5), содержится существенная, с точки зрения физики, неточность, которая может привести к формализму в знаниях школьников. При доведении воды до температуры кипения начинается процесс парообразования при постоянстве последней. Поэтому невозможно дать точный ответ на вопрос задачи, если не указать, какое количество воды превратилось в пар.

При использовании рассмотренного алгоритма на занятиях по информатике следует обратить внимание учащихся на то, что он позволяет правильно решить задачу только в том случае, когда ни одна из темпе-

ратур t_1 и t_2 не равна температуре плавления или кипения. Если же такое равенство имеется, то необходимо уточнить, какое количество вещества переходит из одного агрегатного состояния в другое, и рассчитать необходимое для этого количество теплоты. После указанных уточнений целесообразно предложить учащимся самостоятельно внести в алгоритм и программу соответствующие поправки, что способствует не только более глубокому усвоению школьниками внутренней логики решения физических задач данного типа, но и развитию у них навыков разработки алгоритмов и редактирования программ не на формальном уровне.

А. КАРПЕНКО
Мурманский ГПИ

Как бы есть

Я — простой учитель информатики из сельской глубинки, регулярно читаю ваш журнал. Не всё мне в нем понятно (по образованию я учитель физики, информатику постигал самостоятельно), не всё нравится, но в целом журнал неплохой. Я, конечно, рассуждаю со своей колокольни, но некоторые статьи кажутся мне далекими от жизни, а вот некоторые попадают не в бровь, а в глаз, например «Извините, Елена Константиновна» (№ 2 за 1990 г.). Получила наша школа КВУ УКНЦ в начале августа 1990 г. Я, конечно, обрадовался, прочитав рекламу (в № 1 за 1988 г.) этого комплекса. С трудом нашли организацию, согласившуюся установить КВУ с гарантией на обслуживание. При стоимости комплекса 36 тыс. рублей взяли они за установку 5890 рублей (не многовато ли за соединение машин двумя проводами локальной сети и присоединение двумя проводами к силовому щиту?), включили машины, загрузили, благо с первого включения все работало, и уехали, а точнее, исчезли. А дальше... Дальше остался один на один с этой техникой, которая работает по закону «вроде есть и вроде нет». Судите сами.

Машины вроде есть, но все держится на дисководе. Выйдет он из строя, и пиши пропало, а это не мудрено в условиях школы. Нужен хотя бы встроенный Бейсик.

Вентиляция машин естественная, опять же вроде есть. Вскоре после начала работы начинаются перегревы, прекращается реагирование на клавиши. Сперва таких машин не было, потом стало три. Сейчас три машины стоят на подставках и без верхних крышек. При температуре в классе 17—18 °С этого достаточно, а при 23° все машины встают через 45—50 мин работы. С каждым месяцем все сильнее «дребезг» клавиатуры. Что же будет к концу учебного года?

Программное обеспечение вроде бы есть. К машинам приложен комплект литературы из 11 книг, два диска с ФОДОСом, два диска с КДП. Но из книг по три штуки — копии (зачем?), и многое написано в расчете на профессионального программиста. Коварен сетевой Бейсик: первая же ошибка ученика — и РМУ зависает намертво. И я пол-урока загружаю машины. (Это уже потом я узнал, что сетевой Бейсик работает несколько лучше, если на РМП работает программа связи в режиме «продолжение».) А ведь дисковый Бейсик на РМП работает нормально! Кто, зачем и как это сделал? Почему нельзя передавать по сети дисковый

Бейсик? И кто придумал, чтобы инициатором обмена было РМУ? Подавляющее большинство учащихся простые программы с трудом набирают, а авторы комплекса хотят, чтобы они свои программы на диск записывали. Кстати, иногда при попытке обмена РМУ с РМП файл на диске портится: каталог машина читает, а файл прочесть не может, хотя он может читаться с РМУ. Прикладного же программного обеспечения вообще нет, а как без него воплощать в жизнь рекомендации важных столичных методистов?

Принтер вроде есть, но, чтобы начать на нем печатать, необходимо прогнать тест УКНЦ, а это 7—8 мин.

Итог — я сам со своим классом. Вроде есть, потому что работаю, стараюсь что-то сделать, набираю и записываю простенькие программы на диск (хорошо хоть РМП худо-бедно работает), на большее-то у меня ни времени, ни знаний не хватает. И в то же время меня вроде и нет. Монтажников я больше не интересую, они свой куш уже сорвали, а насчет сервисного обслуживания ответ у них один — ЗИПа нет. Что им 193 рубля в месяц, когда они в неделю делают 6000 рублей на установке! И надежд на ремонт двух (пока) вставших машин нет.

Заводу я не интересен, хотя писал туда два раза: ведь машины и так берут, с руками отрывают, поэтому безбедная жизнь обеспечена, и мне в ответ полное молчание. Писал я и по тем адресам, что рекламировались в вашем журнале. «Логос» вообще не ответил, из Ангарска прислали каталог с космическими ценами, например: RT-11 за 600 рублей.

Вы спросите: чего же ты хочешь? Да ничего. Просто наболело, вот и написал. Можете сразу выбросить мое письмо. Только детей жалко. Ведь каждый урок дискредитирует в их глазах и компьютеры, и информатику. Они уже сейчас нервно вздрагивают при каждом «писке» (РМУ подает сигнал при зависании). И перспективы мрачные: всю говорят, что на следующий год информатика как самостоятельный предмет исчезнет из учебных планов. Зачем же было огород городить, вводить людей в заблуждение? И что прикажете нам делать? Учиться преподавать математику? А сколько раз можно начинать все заново?

Ну, а вам я желаю успехов в работе, всего наилучшего. И ближе к жизни! Журнал ваш прежде всего нужен школе.

В. ГОЛОВНИН
СШ № 1 с. Бакалы Башкирской ССР

Вниманию работников учебных заведений и вычислительных центров!

Предлагаем для оборудования дисплейных классов, рабочих мест операторов ЭВМ и автоматизированных рабочих мест на базе различных ПЭВМ специализированные столы («Стол-АРМ»).

Вам могут быть предложены также услуги по оборудованию кабинетов информатики на базе КУВТ «Сура» (реклама в «ИНФО», № 3 за 1990 г.), «Корвет», «Поиск» и автоматизированных рабочих мест на базе отечественных и импортных компьютеров. Фирма выполнит для вас заказы на поставку оборудования, аудио- и видеотехники.

Получить справки о наличии столов, оборудования, размещении заказов можно по адресу: 310052, Харьков, ул. Котлова, 72 — либо по телефону: 22-86-13 (код 0-572).

Вниманию руководителей учебных заведений, вузов, ИУУ и центров информатики

Алтайский краевой институт усовершенствования учителей совместно с научно-техническим предприятием «Альтернатива» НИИ ИВТ АПН СССР приглашает учителей и методистов пройти обучение на хозрасчетных курсах по использованию в учебном процессе терминальных классов КУВТ-86 (любых типов), УКНЦ-01 (БК-0011), УКНЦ («Электроника МС 0511»), «Корвет» (любых выпусков), классов на основе БК-0010 с магнитофонной сетью.

В программе курсов:

изучение новейших операционных систем, пересылаемых по сети классов; изучение новейших программных разработок НТП «Альтернатива» (Турбо-Паскаль, Турбо-Пролог, Си, Форт, система «Автор» и т. д.); копирование программных пакетов на диски слушателей.

Стоимость обучения, включая стоимость передаваемых слушателям программных пакетов, от 850 до 1000 рублей по перечислению.

Обратите внимание!

Получить программные пакеты в комплекте с обучением значительно выгоднее, чем просто купить пакет!!

Заявки направлять по адресу: 656099, Барнаул, пр. Социалистический, 60, каб. 15. Тел.: (8-385-2) 25-58-24, 23-48-24.

Малое предприятие «Ширакаци»

при Ереванском заводе «Массив»

начало серийный выпуск контроллера накопителя на гибких магнитных дисках 133 мм для ПЭВМ «Электроника БК-0010-01».

Контроллер подключается к МПИ компьютера и аппаратных доработок не требует, выполняет следующие команды: форматирование, инициализация файловой структуры, чтение, запись, переименование, удаление файла.

Контроллер обеспечивает работу двух дисководов и магнитофона в мониторе и в Бейсике.

Всех заинтересованных просим обращаться по адресу: 375079, Ереван-79, МП «Ширакаци». Тел.: 63-24-31.

Выставка программных средств

В декабре — январе 1991 г. на ВДНХ СССР в павильоне «Народное образование» проходила тематическая выставка «Новые педагогические программные средства для средней школы».

На выставке были представлены 324 программных средства учебного назначения, в том числе (см. табл. 1):

отдельные ППС, пакеты ППС, обучающие программно-методические комплексы и системы, предназначенные для поддержки изучения курса информатики и других общеобразовательных предметов;

инструментальные программные средства для создания ППС или их фрагментов;

программы для ведения делопроизводства учителем, директором школы;

компиляторы;

сервисные программы;

игры для детей;

тесты (психодиагностика, профессиональная пригодность и др.).

На выставке функционировали как отдельные ПЭВМ, так и КУВТы: IBM —

совместимые (PS/2, PC/XT, PC/AT), КУВТ «Ямаха» (MSX-1, MSX-2), КУВТ-86 (БК-0010), КУВТ УКНЦ, КУВТ «Корвет», ПЭВМ «Агат», семейство ДВК.

Регионы страны и организации, представившие программные средства на выставку, приведены в табл. 2.

Экспертиза программ проводилась Всесоюзной экспертной комиссией по оценке качества ППС, которая отбирала экспонаты и производила оценку программ, претендующих на медали ВДНХ СССР.

Экспертиза осуществлялась по нескольким направлениям: по типу программного средства, по типу ПЭВМ и КУВТ с учетом различий программно-аппаратного характера, педагогических, дидактических, физиолого-гигиенических требований. Учитывалось также то обстоятельство, что использование программ учебного назначения должно обеспечивать: визуализацию на экране ПЭВМ изучаемых закономерностей, явлений, процессов (в виде графиков, моделей, таблиц, диаграмм), незамедлительную обратную

Таблица 1

Назначение программ	Количество программных средств							
	IBM	«Ямаха»	УКНЦ	«Корвет»	«Агат»	КУВТ-86	ДВК	Итого
Курс ОИВТ	6	22	8	7	3	9	5	60
Математика	12	11	—	—	—	3	—	26
Физика	33	11	3	—	—	11	4	62
Химия	2	3	—	1	1	5	5	17
Русский язык	2	8	3	2	1	1	2	19
Иностранный язык	2	6	—	—	—	—	—	8
Психология	—	3	—	1	1	2	—	7
История	—	—	2	—	—	1	—	3
Экология	—	—	—	—	—	1	—	1
Астрономия	—	1	—	—	—	—	—	1
Рисование	—	1	—	—	—	—	—	1
Экономика	1	—	—	—	—	—	—	1
Инструментальные средства	20	16	11	22	1	23	—	93
Игры и спецдисциплины	—	—	1	11	1	—	12	25
Всего	78	82	28	44	8	56	28	324

Регионы страны, представившие ПС на выставку

Республики	Города	Организации
РСФСР	Москва	Исслед. центр Гособразования СССР, АСП «Кудиц», НЦ ПСО, НИИ СОиУК АПН СССР, МИФИ, МГУ, НИИ ОПП АПН СССР, НИИ школ МО РСФСР, НИИ СЧЕТМАШ, школа № 542, МИИГА, ИПИ АН СССР, НЦ «Фактория»
	Ленинград	Школа № 307
	Новосибирск	НИИ ИВТ АПН СССР
	Казань	Госуниверситет, НПО ГКВТИ СССР
	Свердловск	Пединститут
	Ярославль	ИУУ, пединститут
	Саратов	НПО «Титан»
	Пермь	Центр компьютерного обучения
	Орджоникидзе	Дом техники
	Симферополь	Госуниверситет
	Иркутск	Сиб. отд. АН СССР
	Н. Новгород	Госуниверситет
	Барнаул	ИУУ
	Бердянск	ИПИ АН СССР (филиал)
	Украина	Киев
Херсон	Пединститут	
Запорожье	ПО «Искра»	
Ужгород	Госуниверситет	
Тернополь	Пединститут	
Харьков	Пединститут	
Луганск	Школьный учитель (В. Л. Еськов)	
Донецк	ПТУ связи № 27	
Белоруссия	Минск	Госуниверситет
Пинск	ВТК «Квант»	
Модырь	Школьный учитель (Шведовский)	
Луцк	Пединститут	
Гродно	Госуниверситет	
Брест	ПТУ	
Молдова	Кишинев	Школа-лицей 62
Латвия	Рига	Госуниверситет
	Даугавпилс	Пединститут

связь между пользователем и программой, работу в интерактивном режиме, т. е. предполагающем взаимодействие с программной системой, возможность управления отображением на экране моделей изучаемых объектов, процессов, явлений, динамическое представление информации (при необходимости и педагогической целесообразности).

Результаты экспертизы позволяют утверждать, что за два года, истекших со времени окончания работы последней выставки программных средств на ВДНХ СССР, был сделан существенный позитивный сдвиг (как в качественном, так и в количественном отношении) в области их разработки и использования в целях интенсификации процесса обучения, развития информационной культуры, подготовки подрастающего поколения к жизни и деятельности в период информатизации общества.

Наиболее продвинутым направлением можно считать разработку программного обеспечения курса информатики. Программы,

предназначенные для поддержки других общеобразовательных курсов, пока остаются в целом на достаточно низком уровне.

В составе разработчиков программных средств, представленных к наградам ВДНХ СССР:

учителя, коллективы учителей и учеников	—10
коллективы сотрудников НИИ АПН СССР	—9
коллективы сотрудников вузов и госуниверситетов системы АН СССР	—8
коллективы педагогических вузов и учреждений народного образования	—23

Распределение дипломов и медалей ВДНХ СССР по учебным заведениям и организациям системы народного образования

1. Московский инженерно-физический институт Гособразования СССР за разработку комплекса программных средств, предназначенных для углубленного изучения курса информатики, английского языка, использования во внеклассной и круж-

ковой работе в средних учебных заведениях на КУВТ «Ямаха» и IBM PS/2:
J-LISP — золотая медаль и диплом I степени;
серебряная медаль — 4.

2. Научный центр программных средств Московского городского комитета по народному образованию за разработку пакетов программных средств, предназначенных для поддержки преподавания курса информатики и вычислительной техники, игровых программ, используемых во внеклассной и кружковой работе на КУВТ УКНЦ «Ямаха», IBM PC и ПЭВМ «Агат»:

DISPLAYLAND I. E. DESIGNER —
золотая медаль и диплом I степени;
серебряная медаль — 6.

3. Научно-исследовательский институт средств обучения и учебной книги АПН СССР за разработку обучающей программно-методической системы «Многогранники», предназначенной для формирования основных стереометрических понятий школьного курса геометрии и развития пространственного мышления; за разработку пакетов программных средств поддержки процесса преподавания физики, алгебры, русского языка, а также за разработку пакета программных средств для автоматизации процесса ведения делопроизводства в средних учебных заведениях на PS/2 и КУВТ «Ямаха»:

Обучающая программно-методическая система «Многогранники» —
золотая медаль и диплом I степени;
серебряная медаль — 4.

4. Научно-исследовательский институт счетного машиностроения за разработку пакетов программных средств, предназначенных для поддержки процесса преподавания курса информатики на КУВТ «Корвет»:

серебряная медаль — 3.

5. Институт проблем информатики АН СССР за разработку пакетов инструментальных программных средств, предназначенных для конструирования прикладных программ учебного назначения, программно-обучающей среды, а также поддержки процесса преподавания курса информатики и вычислительной техники на КУВТ IBM PC и «Корвет»:

серебряная медаль — 3.

6. Научно-технический и потребительский кооператив «ИнПроф» за разработку программных средств, предназначенных для поддержки преподавания курса информатики на КУВТ «Ямаха»:

серебряная медаль — 1.

7. Компьютерный учебно-демонстрационный и информационно-издательский сервисный центр «КУДИЦ» за разработку программно-методического средства «НИТ в образовании» и пакетов программных средств, предназначенных для подготовки учителей-преподавателей информатики, а также пакетов программных средств, предназначенных для поддержки преподавания курса физики, английского языка на КУВТ IBM PS/2:

Программно-методический комплекс «НИТ в образовании» —
золотая медаль и диплом I степени;
серебряная медаль — 4.

8. Малое промышленное предприятие «ИРЕЛА» за разработку пакета инструментальных

программных средств, предназначенного для разработки обучающих программ на IBM PC:

серебряная медаль — 1.

9. Научно-производственный кооператив «Кортеж» за разработку пакета программных средств, предназначенного для автоматизации процесса ведения делопроизводства на IBM PC:

серебряная медаль — 1.

10. Совместное научно-производственное предприятие «Континент» за разработку пакета программных средств, предназначенного для углубленного изучения физики в средних учебных заведениях на основе моделирования изучаемых процессов и явлений на КУВТ «Ямаха»:

серебряная медаль — 1.

11. Кооператив «Полином» за разработку пакетов программных средств, учебно-методических комплексов, пакетов инструментальных программных средств и средств программно-аппаратной поддержки процесса преподавания на КУВТ «Корвет», УКНЦ, «Ямаха» курса информатики, физики, математики и других общеобразовательных предметов:

Редактор контролирующих программ —
золотая медаль и диплом I степени;
серебряная медаль — 8.

12. Ассоциация педагогов и психологов «Новые технологии обучения и развития» за разработку пакета программных средств, предназначенного для углубленного изучения отдельных тем курса физики на IBM PC:

серебряная медаль — 1.

13. Совместное советско-американское предприятие «Диалог» за разработку пакета инструментальных программных средств, предназначенного для разработки программ автоматизированного обучения и контроля знаний на IBM PC:

серебряная медаль — 1.

14. Свердловский государственный педагогический институт за разработку пакета программных средств, предназначенного для изучения отдельных тем курса информатики на IBM PC и программно-методического комплекса по ОИВТ на КУВТ «Ямаха»:

серебряная медаль — 2.

15. Симферопольский государственный университет за разработку пакетов программных средств, предназначенных для поддержки процесса преподавания отдельных тем курса русского языка на IBM PC и КУВТ «Ямаха»:

серебряная медаль — 1.

16. Минский государственный университет БССР за разработку пакета программных средств, предназначенного для формирования орфографических умений и навыков при изучении русского языка на базе систематизированных и обобщающих правил на КУВТ «Ямаха»:

серебряная медаль — 1.

17. Харьковский государственный педагогический институт за разработку пакета программных средств, предназначенного для обучения языкам программирования и пакета программных средств, предназначенного для контроля знаний учащихся на КУВТ «Ямаха»:

серебряная медаль — 2.

18. Тернопольский государственный педагогический институт за разработку инструментального программного средства, предназначенного для расширения возможностей графики при изучении ОИВТ и организации самостоятельного изучения учебного материала на КУВТ «Ямаха»:

серебряная медаль — 1.

19. Научно-исследовательский институт педагогики УССР за разработку педагогического программного средства, предназначенного для построения и демонстрации исполнения блок-схем учебных алгоритмов на КУВТ «Ямаха»:

серебряная медаль — 1.

20. Институт математики и информатики Латвийского университета за разработку программно-аппаратной поддержки функционирования пакетов программных средств автоматизации создания обучающих программ на КУВТ «Ямаха»:

серебряная медаль — 4.

21. Луцкий педагогический институт за разработку программных средств системы оперативного контроля уровня знаний на КУВТ «Ямаха»:

серебряная медаль — 1.

22. ПТУ № 27 г. Бреста за разработку пакетов программных средств профессионального пользования по предмету «Правила технической эксплуатации на железных дорогах СССР» на КУВТ «Ямаха»:

серебряная медаль — 1.

23. Научно-исследовательский институт школ РСФСР за разработку пакета программных средств досугового назначения на ПЭВМ «Агат», предназначенного для развития логического и пространственного образа мышления, памяти:

серебряная медаль — 1.

24. Союз научных и инженерных обществ СССР, Физическое общество СССР за разработку пакетов программных средств и комплекса сервисных программ языка Си, Паскаль на КУВТ «Корвет»:

серебряная медаль — 2.

25. Школа № 307 Ленинграда за разработку комплекса педагогических программных средств информационного, моделирующего, контролирующего назначения по физике, истории и психолого-педагогических тестов на КУВТ-86:

серебряная медаль — 1.

26. Нижегородский государственный университет им. Н. И. Лобачевского за разработку пакетов прикладных программ тренировочного, контролирующего, игрового типа, предназначенных для изучения трудноусваиваемых тем курса физики, химии на КУВТ-86:

серебряная медаль — 1.

27. Ярославский государственный педагогический институт им. К. Д. Ушинского за разработку пакета компьютерных дидактических материалов по русскому языку, предназначенного для проведения контролирующих занятий и самостоятельной работы на КУВТ «Ямаха»:

серебряная медаль — 1.

28. Научно-исследовательский институт информатики и вычислительной техники АПН СССР за разработку программно-методического комплекса «Учитесь читать», предназначенного для обу-

чения учащихся младших классов навыкам чтения без использования речедвигательного компонента на КУВТ «Ямаха»:

серебряная медаль — 1.

29. Гродненский государственный университет за разработку пакета программных средств психолого-педагогического тестирования профессиональной пригодности на учительскую профессию на КУВТ «Ямаха»:

серебряная медаль — 1.

30. «Узбекучавтоматика» г. Ташкента за разработку пакета педагогических программных средств, предназначенного для самостоятельного изучения правил построения предложений в узбекском языке с возможностью синхронного прослушивания слов и конструкций на КУВТ «Ямаха»:

серебряная медаль — 1.

31. Производственное объединение «Искра» г. Запорожья за разработку аппаратно-программных средств поддержки сети ПЭВМ «Агат», предназначенных для объединения в сеть до 15 ПЭВМ с возможностью расширения объема памяти и скорости обмена:

серебряная медаль — 1.

32. Психолого-педагогический кооператив «Второе дыхание» за разработку пакета программных средств, предназначенного для поддержки процесса преподавания курса русского языка на ПЭВМ «Агат»:

серебряная медаль — 1.

33. Средняя школа г. Луганска за разработку пакета программных средств, предназначенного для проведения уроков по алгебре и началам анализа для X классов:

серебряная медаль — 1.

В ряду мероприятий, организованных в рамках выставки — два Всесоюзных семинара разработчиков педагогических программных средств ППС и представителей республиканских ИУУ «Новые информационные технологии в системе общего среднего образования» (26—28 декабря 1990 г.) и учителей информатики «Педагогические аспекты использования новых информационных технологий в системе общего среднего образования» (14—16 января 1991 г.).

В заключение суммируем многочисленные отзывы о прошедшей выставке и предложения по дальнейшей организации подобных выставок и семинаров.

1. В целом устные и письменные отзывы участников семинаров и посетителей выставки сходятся в одном: выставка полезна, содействует обмену передовым опытом разработки ППС и их использования в учебном процессе в общеобразовательных школах, способствует повышению квалификации и разработчиков, и пользователей, учителей-практиков.

2. На выставке были полностью представлены различные виды учебной вычисли-

тельной техники, используемой в школах, что является положительным фактором.

3. Большой интерес вызвали экспериментальные проекты «Пилотные школы» и «Школьная электронная почта».

4. Предлагается организация аналогичных республиканских, региональных, краевых выставок, предшествующих всесоюзной.

5. Тематика семинаров представляла интерес для разных категорий слушателей, поэтому было предложено организовывать подобные семинары для объединенной аудитории разработчиков и учителей-практиков.

6. Встреча учителей информатики с учеными-психологами, сотрудниками НИИ общей и педагогической психологии АПН СССР, в рамках второго семинара, объединенная общей темой «Психолого-педагогические проблемы использования НИТ в преподавании школьных дисциплин», вылилась в активный обмен мнениями, вышедший за рамки объявленной темы.

7. Коммерческий характер выставки вызвал двойное отношение: с одной стороны он дает стимул разработчикам и ориентирует пользователей и руководящих работников в стоимости разработок ППС для учреждений общего среднего образования, с другой — в школах отсутствуют средства на приобретение необходимых разработок. Несмотря на то, что в период проведения выставки фактически не подписано ни одного договора на продажу ППС, представляющие их фирмы надеются на изменение ситуации в дальнейшем.

Обеспечение школ программными средствами, получившими медали, производится централизованным порядком. Заявки направлять через региональные магазины учебно-наглядных пособий «Учколлектор» в Казанский производственный комбинат программных средств.

123

Обращение к учителям информатики

Ассоциация учителей информатики обращается к учителям, которые ведут опытно-экспериментальную работу в области школьной информатики и межпредметных связей.

Основные направления опытно-экспериментальной работы, интересующие ассоциацию:

компьютер в начальной школе;
курс информатики в V—VII классах;
курс информатики в VIII—IX классах;
курс информатики в X—XI классах;
кружковая, факультативная, клубная работа по информатике;

использование компьютеров на уроках физико-математического, естественного, гуманитарного циклов, интегрированные уроки и курсы;

классы с углубленным изучением информатики;

классы с углубленным изучением программирования;

работа со способными детьми.

Желающих наладить контакт с ассоциацией просим ответить на прилагаемую анкету. Полученная информация станет основой для решения следующих вопросов:

придание особо интересным работам в области школьной информатики статуса экспериментальной площадки ассоциации;
приоритетное выделение ВТ под опытно-экспериментальные площадки;

помощь в приобретении учебников, методических пособий, программного обеспечения;

организация и проведение научно-мето-

дических семинаров по каждому из выделенных направлений;

научное руководство опытно-экспериментальной работой.

Адрес для переписки: 456321, Челябинская область, г. Миасс, пр. Октября, 25, школа № 18, Алексеевой Т. М.

Анкета для участников опытно-экспериментальной работы:

1. Фамилия, имя, отчество, возраст, почтовый адрес, телефоны (домашний, рабочий, код).

2. Место работы, квалификация, стаж, общая характеристика опыта работы.

3. Направление или тема эксперимента (опытной работы).

4. Краткое содержание эксперимента (опытной работы).

5. Временные характеристики эксперимента (опытной работы).

6. Результаты эксперимента (опытной работы).

7. Есть ли научный руководитель?

8. Зарегистрирован ли эксперимент (опытная работа) в отделах народного образования?

9. Оказывают ли отделы народного образования материальную и моральную поддержку?

10. Знаете ли адреса подобных экспериментов, поддерживаете ли с ними связь?

11. Какая помощь необходима для успешного ведения опытно-экспериментальной работы?

Наука, искусство и компьютеры

В конце февраля в Москве состоялась советско-американская конференция-симпозиум «ГрафиКон'91» — «Компьютерная графика в науке и искусстве», организованная АН СССР при поддержке и содействии американской администрации и SIGGRAPH — графической группы из Ассоциации по информатике и вычислительной технике (ACM).

SIGGRAPH — одно из крупнейших подразделений ACM. Еще в 1975 г. оно объединяло всего лишь 300 человек, а сейчас в нем более 12 тыс. членов, 54 % которых — химики, художники, архитекторы и т. п. Ежегодно оно устраивает в США свои конференции. На последнюю, SIGGRAPH'90, собралось более 25 тыс. (!) участников. Что же привлекает столь большую аудиторию?

24 Чтобы ответить на этот вопрос, отделим для начала компьютерную графику (КГ) от вещей, которые ею не являются, хотя и похожи. Например, изображение картины Рубенса, оцифрованной сканером, записанное на диск, а затем выведенное на экран монитора — это не то, что принято обозначать термином КГ. В данной процедуре возможности компьютера не используются достаточно существенно, фактически он выполняет роль цифрового телевизора. А вот «ручки-ножки-огуречик», нарисованные на экране с помощью мыши — это уже КГ; компьютер выступает здесь в качестве принципиально нового инструмента художника. Еще более «компьютерны» рисунки, создаваемые написанными на различных алгоритмических языках программами.

Итак, первая, наиболее массовая (хотя и не самая важная) сфера приложения КГ — создание с помощью компьютера киноэффектов, мультфильмов, иллюстраций к архитектурным проектам (по чертежу здания с полной иллюзией натуральности моделируются его внешний вид, интерьер помещений, так что можно осмотреть непостроенный дом со всех сторон и «пройтись» по нему), произведений изобразительного искусства (да, компьютерная живопись уже признается искусством, хотя зачастую совсем не похожа на живопись традиционную) и т. п. Именно эта сфера привлекает к КГ большую часть поклонников и энтузиастов. Действительно, практическая выгодность и широкая применимость подобных приложений налицо. Архитектору становится легче не только проектировать, но и продавать проекты (товар лицом!), кинорежиссеры получают возможность сравнительно дешево использовать для съемок всю вселенную, издатели могут поста-

вить типографию на любой стол... А компьютерная живопись чрезвычайно увлекательна.

График функции, построенный ПЭВМ по таблице или формуле, тоже КГ, хотя и другого типа. Обычно представление числовых данных в виде неких визуальных конструкций называют не компьютерной графикой, а визуализацией. И хотя изображение функции на экране не так эффектно, как «космические» кадры «Звездных войн», и в эту сферу приходят художники. Способность мыслить образами помогает им найти достаточно наглядные, обозримые способы представления огромных массивов числовых данных, получаемых в различных вычислительных экспериментах. В своем исходном виде эти данные практически бесполезны — какой человек разберется в миллионах цифр, характеризующих обтекающие самолет воздушные потоки или (пример, подробно рассматривавшийся на конференции) процесс втекания расплавленной пластмассы в литьевую форму? Только свертка, преобразование в графическую форму делает их доступными для восприятия. Но как изобразить одновременно температуру, давление, направление и скорость движения в каждой точке втекающей в форму пластмассы? Подобные задачи зачастую оказываются более сложными, чем проведение самого вычислительного эксперимента*. Зато, когда приходит успех, получившееся визуальное представление чудесным образом открывает перед учеными тонкие, ранее незаметные и весьма существенные закономерности процессов. Фактически художник, ученый и компьютер становятся единым мыслителем (заинтересовавшись этим аспектом визуализации рекомендуем прочесть интервью с А. А. Зенкиным, одним из первых советских энтузиастов такого подхода, в «ИНФО» № 6 за 1989 г. — «Экология мышления»).

Третий аспект КГ — ее влияние на развитие вычислительной техники. Сложность графических задач стимулирует и увеличение вычислительных мощностей, и повышение возможностей устройств ввода и вывода информации. Графические рабочие станции обычно ненамного отстают от суперкомпьютеров и во многих отношениях находятся на самом переднем крае компьютерного прогресса.

Особо — last but not least — отметим роль КГ в образовании, которой была посвя-

* Статья М. Стернина «Тысяча лиц» о весьма оригинальном методе визуализации многомерных данных опубликована в «ИНФО» № 4 за 1988 г. — Примеч. ред.

щена большая часть выступления директора Института прикладной математики им. М. В. Келдыша АН СССР С. П. Курдюмова на открытии «ГрафиКона'91»: «Есть еще один аспект визуализации, компьютерной графики. Это аспект образования. Научные дисциплины очень дифференцировались. Специалист-математик часто не понимает своего коллегу, а что уж говорить об огромных массах людей. И вот появился новый интегрирующий элемент визуализации знаний, элемент нового способа обучения.

Как утверждают социологи, время внедрения нового знания в тело культуры — 25—30 лет. Ведь нужно переподготовить преподавателей школ и колледжей, нужно переподготовить преподавателей вузов и пропустить школьников и студентов через этот штат переподготовленных людей. А новое знание производится со все возрастающей скоростью. Мало того — оно устаревает наполовину за 5—7 лет, в основном в наиболее передовых областях. Этот разрыв все увеличивается и увеличивается, и необходимы новые средства передачи знания, непрерывного самообразования, непрерывной учебы всех, в том числе преподавателей и ученых. И именно визуализация, образность, связь науки, культуры с искусством — это то могучее средство, которое позволит быстрее сделать доступным новое знание для огромной массы людей».

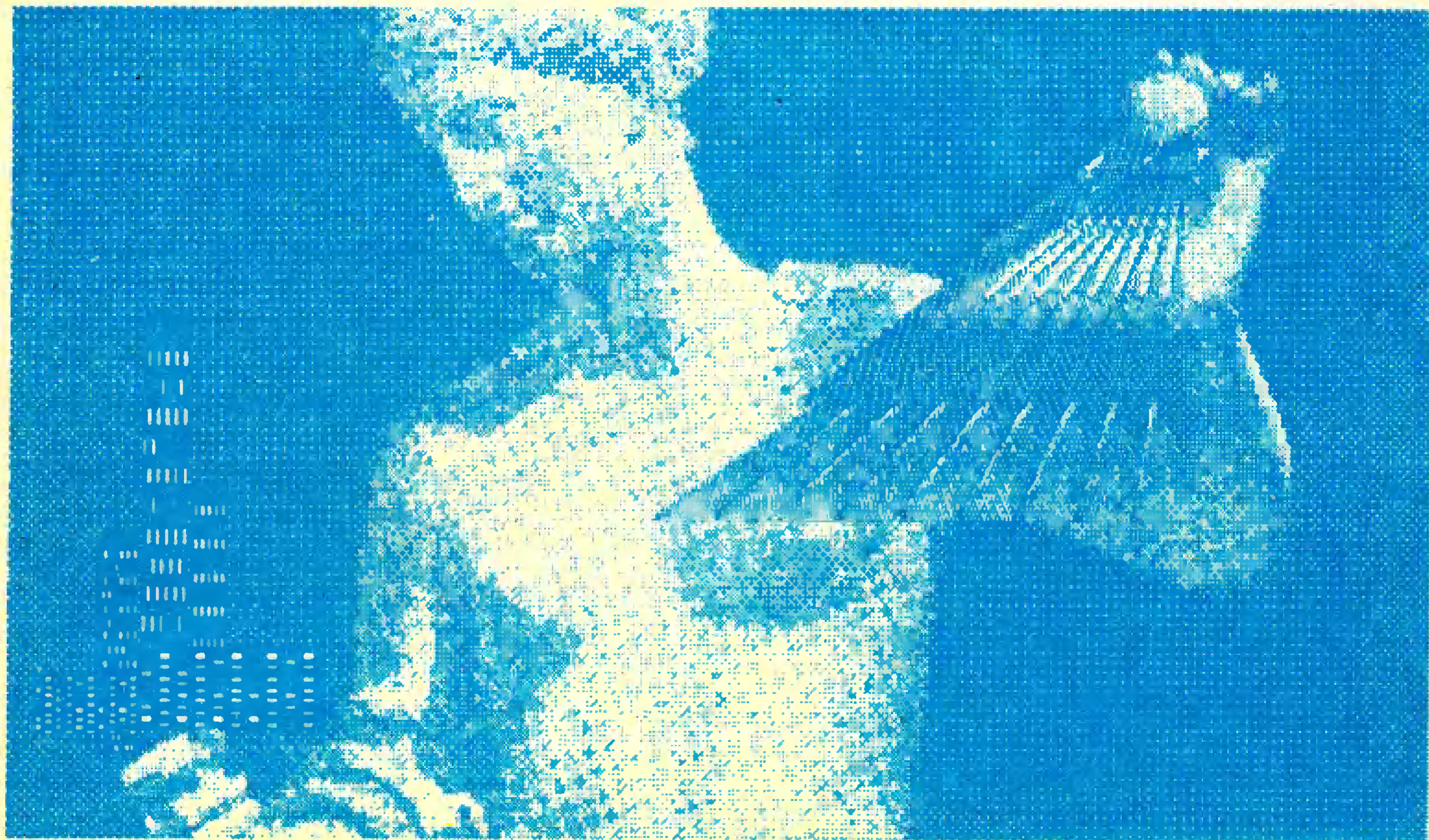
В США компьютерная графика является наиболее широко и динамично развивающимся приложением вычислительной техники. В СССР дело обстоит несколько хуже. Благодаря смягчению позиции КОКОМ у нас в продаже появилась графическая станция

«американского» стандарта «Беста» (фото на первой странице обложки), но ведь нужно еще научиться эффективно ее использовать! Для решения подобных проблем, непосредственного обмена опытом, привлечения новых сил в данную область и была организована конференция-симпозиум. Чрезвычайная широта и значительная порой сложность обсуждавшихся на ней вопросов не позволяют изложить их достаточно подробно. Некоторое представление об узловых моментах дадут цитаты из материалов конференции. *Дон Гринберг, профессор машинной графики, директор Программы машинной графики в Корнелльском университете. Из выступления на открытии конференции.*

...Компьютерная графика — нечто большее, чем просто изображение, созданное машиной. Компьютерная графика для меня — глобальный, универсальный язык, с помощью которого будут общаться человек и компьютер, не нуждаясь в переводчике. Как говорится, лучше один раз увидеть, чем 1024 раза услышать.

Майкл Бейли, руководитель направления научной визуализации в Суперкомпьютерном центре Сан-Диего, профессор Калифорнийского университета по прикладной математике и инженерным наукам. «Научная визуализация для большого, пространственно разобленного коллектива пользователей».

...Обычно при обсуждении научной визуализации упоминается фактор «а-га!» — тот момент, когда научные данные удается представить в таком виде, что приходит новое понимание проблемы. Лучшим в таком процессе познания является то, что на этом



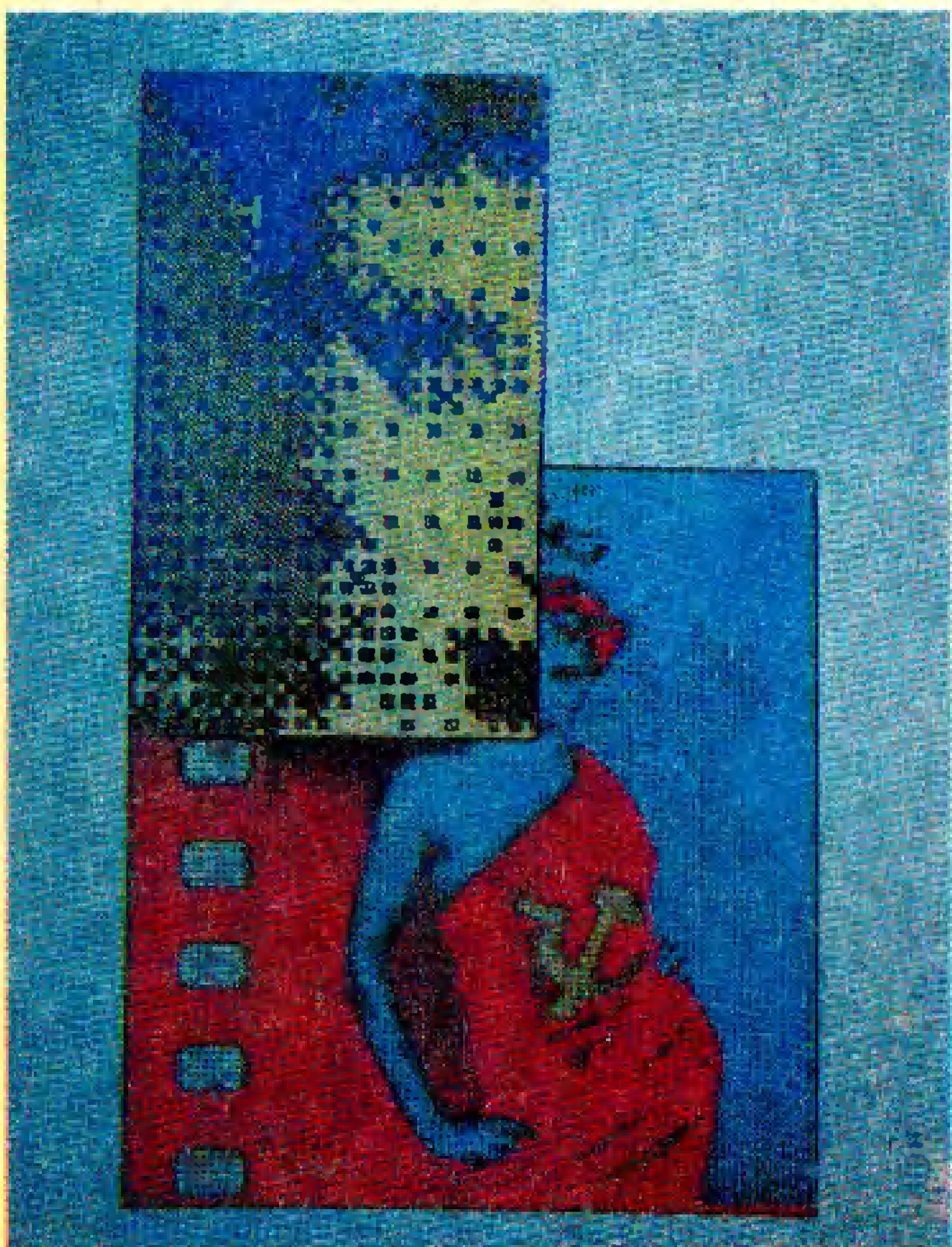
он не заканчивается. То «а-га!» ведет к другим вопросам, которые затем приводят к новому пониманию и т. д.

Джеймс Кларк, основатель и председатель компании «Silicon Graphics, Inc.», доктор информатики, в прошлом профессор Стэнфордского университета по информатике. «Компьютерная визуализация в 80-х и 90-х».

...Исследования по проектированию интегральных схем, в частности, разработки СБИС, финансировало министерство обороны США, рассчитывая тем самым обеспечить конкурентоспособность США на мировом рынке. ...

В конце 1981 г. я безуспешно пытался продать лицензию на Геометрическую Машину Hewlett Packard, IBM, Apollo Computer, DEC и другим графическим компаниям, которые тогда существовали. Однако многие не верили, что трехмерная компьютерная графика важна или что «академические» ученые могут довести до конца какую-либо разработку. Тогда мы с моей командой решили в начале 1982 г. основать фирму «Silicon Graphics». ...

Наши первые покупатели были из кино и развлекательной индустрии. Они были также в числе наиболее требовательных клиентов, поскольку хотели высшего уровня реализма изображения для телевизионных рекламных передач и для развлекательных фильмов. Фактически все специальные эффекты, которые мы видим сегодня по телевидению или в кино, произведены с помощью нашего оборудования. ...



На сегодня мы имеем машину, которая вскоре будет выпущена на рынок с ценой менее 10 тыс. долларов. Ее центральный процессор может выполнять 30 млн машинных операций в секунду; она может строить 15 млн трехмерных векторов и 1 млн окрашенных многоугольных граней в секунду. ...Наиболее перспективная система из тех, что мы сегодня выпускаем, имеет восемь процессоров суммарной производительностью 240 млн машинных команд и 32 млн операций с плавающей точкой в секунду. *Донна Кокс, помощник директора образовательных программ в Национальном центре суперкомпьютерных приложений, доцент Школы искусства и дизайна в Иллинойском университете. «Взаимопроникновение искусства и науки в суперкомпьютерной графике».*

Искусство научной визуализации. ...Первостепенная цель вычислительной науки, питающая быстро растущее научное направление — не просто имитировать оптически воспринимаемую человеком реальность, а дать зрительный образ моделей, которые точно и полно описывают физические системы.

Научная визуализация — это процесс организации и представления числовых данных в виде значимых образов. Она позволяет ученым строить модели абстрактных объемов данных таким образом, что их можно «увидеть» и тем самым быстрее осмыслить. Большой вклад в эту оформляющуюся область деятельности могут внести художники. ...

В XV в. философы Возрождения лелеяли веру, что исследование природы через зрительные образы может выявить ее скрытые законы. Сегодня задача научной визуализации в том, чтобы породить образы недоступных глазу феноменов и тем самым способствовать продолжению процесса познания. У талантливых художников, работающих в области визуализации, есть природный дар представлять абстрактные понятия в виде осмысленных картинок. ...

Художники в процессе своего обучения, как правило, не бывают подготовлены к тому, чтобы использовать современные технологии в качестве средств творчества. Система образования не дает студентам гуманитарных дисциплин даже элементарных знаний в области логики, технологий и точных наук. Поэтому художники, склонные к творческим поискам за пределами традиционных рамок, часто ощущают отсутствие нужных элементов профессиональной подготовки. ...

«Ренессансные группы». ...В настоящее время работа в творческой группе — это тот подход, который позволяет художнику экспериментировать в области научной визуализации.

зации и определять ее направление, используя современную технологию. «Ренессансной группой» называют группу специалистов с дополняющими друг друга сферами компетентности, которые работают сообща, соединяя свои возможности в анализе, технологиях и художественном творчестве, чтобы расширить поле вариантов при решении проблемы. Вклад художников в этот процесс может быть весьма ценным, так как у них заранее развита способность визуально организовывать информацию.

Эд Кэтмулл, Председатель Совета и главный научный руководитель фирмы «Pixar». «Компьютерная анимация для кино и телевидения».

...Принципиальная трудность перепоручения компьютеру «черновой», промежуточной работы при рисовании кадров мультфильма заключается в том, что рисунки художника — это плоские проекции объемного персонажа, причем такие, какими их представляет себе художник; следовательно, часть информации в них потеряна. Например, одна нога закрывает другую. Эта потеря информации существенно затрудняет возможность автоматического получения промежуточных фаз. *Дон Гринберг. «Использование компьютерной графики в архитектурном проектировании».*

...Имеющееся сегодня программное обеспечение в действительности недостаточно подходит для итеративных процессов. Большинство систем первоначально были предназначены для автоматизации процесса черчения. Их можно использовать для обозначения осей, расстановки размеров и нанесения большого числа деталей. Они способны

быстро и экономно чертить в двухмерном пространстве планы, разрезы, фасады и особенно удобны для модификации чертежей.

Однако необходимость точного определения параметров, естественная в чертеже и требуемая такими системами, мешает использовать их на этапе создания эскизов — наиболее сложном и творческом в работе архитектора.

...новые методы синтеза позволяют создавать высококачественные фотореалистические изображения. Однако пока еще не ясно, как эти технологические достижения будут применяться в архитектуре и как они повлияют непосредственно на процесс проектирования.

Еще одна цитата из доклада Донны Кокс.

...Скульптор Нэнси Бэрсон и ученые из Массачусетского технологического института совместно разработали методы, позволяющие на цифровой основе «старить» человеческие лица, экстраполируя изменение черт лица в будущее. Среди ранних компьютерных изображений такого рода — юмористические или шокирующие «старческие варианты» лиц Брука Шильдса и Джона Траволта, обрюзгшие и морщинистые. Однако «машина старения» способна также проецировать в будущее детское лицо, благодаря чему она стала ценным инструментом розыска пропавших детей.

* * *

Цитирование можно было бы продолжать еще и еще, но, думается, сказано уже достаточно, чтобы читатели получили представление о целях, задачах и методах компьютерной графики.

Ленинградский производственный кооператив «ОСТ» предлагает

**программное обеспечение для компьютеров ДВК-2М, 3М,
КУВТ-86, УКНЦ.**

Кооператив располагает обширным банком игровых, учебных, системных, прикладных программ.

Расценки значительно ниже государственных.

Списки программ (при указании типа ЭВМ) высылаются бесплатно.

Адрес: 199034, Ленинград, а/я 437.

Четыре миллиона знаков

Для того чтобы закодировать 32 русские и 26 латинских букв, 10 цифр и еще пару десятков разных значков и знаков препинания, компьютеру достаточно 8-разрядного слова. 16-битное слово позволяет закодировать уже более 64 тыс. символов, которых хватит, чтобы загрузиться не только в кодировках, но и в алфавитах, ими представленных. Японская фирма NEC начала продавать переносные компьютеры 10LA II, которые имеют 32-разрядные слова, что позволяет процессору оперировать сразу 4 млн. знаков.

Новый лэптоп ориентирован специально на деловых людей и предназначен в основном для подготовки документации, обмена информацией по телефону и проведения несложных расчетов, с которыми справился бы любой калькулятор. Так для чего японскому бизнесмену мощная 32-разрядная ЭВМ? А хотя бы для того, чтобы закодировать все иероглифы. И вообще, если есть выбор (и деньги), многие предпочтут легкому фиату солидный мерседес.

Товар — лицом

Специалисты в области маркетинга и сбыта электронной продукции отмечают, что доля программистов среди покупателей компьютеров снижается. Все больше компьютеров находят применение в проектировании, делопроизводстве и системах управления.

Видимо, именно это наблюдение заставило японскую фирму Fujitsu поставить на конвейер новую модель настольного персонального компьютера FMR-70 HX2S, который почти не отличается от уже существующих моделей IBM-совместимых ПЭВМ. Но за этим «почти» кроется именно то, что сегодня для массового покупателя наиболее привлекательно: высокоскоростной дисплей с хорошей разрешающей способностью. «Быстрая» и детальная графика нужна всем: и архитекторам, и конструкторам, и полиграфистам. Чтобы удовлетворить их требования, фирма спроектировала высокоскоростной графический адаптер NSGA, позволяющий практически мно-

ЧТО МОЖЕТ ЭВМ?

венно отобразить на экране цветное изображение, состоящее почти из миллиона точек (1120×750).

Все остальные «тактико-технические данные» нового компьютера не уступают качеству графика: 32-разрядная архитектура, i386 центральный процессор, работающий на частоте 20 МГц, два флоппи-дисковода, до 16М байт ОЗУ, встроенный винчестер на 75М байт. И все-таки первый взгляд покупатель бросает на экран.

Микропроцессор или суперкомпьютер?

Толковый словарь по вычислительной технике и информатике, пожалуй, самое нестабильное справочное пособие в наши дни. Быстро развивающаяся технология производства компьютеров нередко существенно меняет смысл терминов.

Вот наглядный пример. Фирмой Intel был выпущен новый процессор 80860. Но по основным параметрам, указанным в словаре, он не уступает суперкомпьютерам. И это не удивительно. На единой подложке новой СВИС расположено по крайней мере три процессора: центральный, процессор для выполнения операций над числами с плавающей запятой и графический процессор для работы с трехмерной графикой. Здесь же расположены два регистровых файла, КЭШ-память для хранения данных и команд и устройство управления оперативной памятью, совместимое с ОЗУ процессора 80386.

Такая конструкционная плотность позволила не только сократить объем компьютера и его стоимость, но и повысить тактовую частоту опорного генератора до 40 МГц. В результате однокорпусный процессор i80860 может конкурировать с существ-

ующими многопроцессорными системами обработки информации.

А для того чтобы называться суперкомпьютером, новому однокристальному процессору не хватает совсем немногого: устройства для ввода и вывода информации в корпусе. Так что то, что сегодня называют суперкомпьютером, завтра может оказаться скромным микропроцессором.

Семь секунд — много или мало?

Если это время разгона автомобиля до скорости 100 км/ч, то, скорее, мало. Мировой рекорд в этой области — порядка 2—3 с. Но у нас-то речь идет о разгоне диска в дисковом!

Магнитооптические носители информации уже вышли из младенческого возраста, но до радикальной победы над своими недостатками им еще далеко. Большой процент брака при записи и считывании сверхплотной информации требует использования специальных алгоритмов исправления ошибок, что, в свою очередь, приводит к увеличению объема файлов и уменьшает быстрейшее действие дисковода.

В частности, в новой модели RMD 5100-S компании Nakamichi Peripherals Corp. (на диск диаметром 3,5 дюйма записывается 128М байт данных) готовность к работе достигается через 7 с после включения; 4 с уходит на раскрутку диска и 3 — на считывание алгоритма исправления ошибок.

Новый «блокнот» «Тошибы»

Он называется T1000LE, весит 2,9 кг, полностью совместим с IBM PC, имеет ОЗУ емкостью 640К байт (фирма обещает в скором времени довести ее до 1М байт). Главная особенность этой модели — два накопителя внешней памяти: жесткий диск емкостью 20М байт и дисковод для 3,5-дюймовых дискет на 1,44М байт. Согласитесь, что разместить их в «блокнотном» компьютере нелегко.

Цена T1000LE — 1699 долларов.



ПРОГРАММНО - ИНСТРУМЕНТАЛЬНАЯ

ОБОЛОЧКА

для разработки программ автоматизированного обучения
или контроля знаний

Версия 1.1

Система может быть использована преподавателями любых учебных заведений для разработки автоматизированного обучения или контроля знаний учащихся практически по любой дисциплине.

Для работы с Системой не предусматривается знание основ программирования. Вся работа по разработке автоматизированного обучения или контроля ведется в диалоговом режиме посредством МЕНЮ.

Система состоит из двух программ: Учитель и Ученик.

Программа Учитель - служит для разработки обучающего или контролирующего урока по заранее написанному сценарию. Автор сценария, по понятным причинам, должен быть опытный педагог. Методика составления сценария подробно описана в "Инструкции Пользователя".

Программа Ученик - воспроизводит на экране монитора весь алгоритм разработанного Учителем сценария обучения или контроля:

- пред"являет обучаемому учебный материал: теоретический, задачи и вопросы;
- контролирует и комментирует действия обучаемого;
- адаптируется к обучаемому : регулирует уровень сложности выдаваемого учебного материала;
- оценивает знания обучаемого.

Система содержит ГРАФИЧЕСКИЙ ПАКЕТ, с помощью которого имеется возможность, наряду с текстовыми ДОЗАМИ обучения-контроля, воспроизводить на экране всевозможные графические иллюстрации (геометрические фигуры, электрические схемы и т.п.).

Система написана на языке "С", функционирует в операционной системе MSDOS. (PCDOS) для версий 3.10 и выше.

Учебный центр филиала СП ДИАЛОГ в г.Наб.Челны ежемесячно проводит обучение работе с Системой. Срок обучения 3-5 дней.

Ведется разработка последующих версий, которые в первую очередь будут поставляться законным пользователям предыдущих версий.

Стоимость версии 1.1:

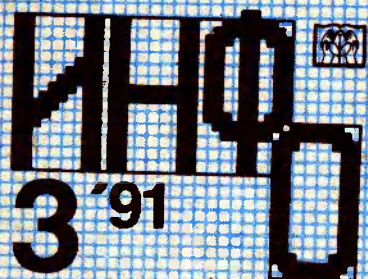
Инструкция пользователя + 2 дискеты - 940 руб.

сп
Диалог

Совместное советско-американское предприятие.
Филиал в г. Набережные Челны
423815, г. Наб. Челны, Шишкинский бульв., 16
Телефон: 54-82-27, 56-14-67,

Цена 1 р. 20 к.
70423

УПК



**ИНФОРМАТИКА
И ОБРАЗОВАНИЕ**

