

ИНФОРМАТИКА И ОБРАЗОВАНИЕ

1 1992



Фирма EXPRESS-CONTACT Co.

представляет

компьютерную систему LECBIS® V.01, предназначенную для обучения грамматике английского языка

Система позволяет:

овладеть навыками чтения и произношения;
значительно пополнить запас английских слов (в системе используется более 8000 наиболее употребляемых слов);
прочно овладеть грамматическими правилами английского языка (около 1400 практических упражнений).

Система характеризуется:

простотой в работе. Рассчитана на пользователя, не имеющего навыков работы с компьютером типа IBM PC;
разнообразием практических упражнений и учебного диалога;
автоматическим доступом к новому материалу по мере усвоения предыдущего;

доступом к теоретическому материалу в процессе обучения и поощрением многократного обращения к нему вместо его заучивания;

возможностью получения каждым учащимся машинописной копии теоретического материала с подробными практическими примерами по каждой изучаемой теме;

средствами адаптации и настройки системы в соответствии с индивидуальным уровнем подготовки обучаемого;

возможностью перенастраивания LECBIS с функций преимущественно обучения на функции контроля и оценки знаний и навыков обучаемого.

Программное обеспечение:

работает на различных конфигурациях персональных компьютеров и их сетей, совместимых с IBM PC/XT/AT/PS-2 в MS-DOS или PS-DOS версий 3.00 и старших с объемом оперативной памяти не менее 256 Кбайт;

работает с монохромными или цветными дисплеями и адаптерами типа CGA, EGA, VGA, Super-VGA в текстовом и графическом режимах;

каждая из 22 обучающих программ, из которых состоит система LECBIS V.01, может работать автономно.

Приобретение:

бесплатное предпродажное обучение пользователей LECBIS V.01;
покупка за рубли как всей системы, так и ее отдельных программ;
гарантийное, в течение 3 лет, и послегарантийное обслуживание;
льготы для школ 40 %;

продажа последующих версий системы с 70 % скидкой;

лица, распространяющие или внедряющие LECBIS, стимулируются фирмой в размере 1500 рублей за каждый реализованный комплект.

Фирма может поставить персональные компьютеры типа IBM PC и их сети по цене от 6000 до 10 000 руб. за один компьютер.

Представительство на Украине:

257022, Черкассы, ул. Горького, д. 6. Тел. (8-0472) 43-33-01.



ИНФОРМАТИКА И ОБРАЗОВАНИЕ

1 '92
январь —
февраль



Содержание

Общие вопросы

Лапчик М. Информатика и технология: компоненты педагогического образования	3
Белошапка В. Информатика как наука о буквах	6
Варданян И. АСУ обучением с минимальными потерями	13

Методика обучения

Гейн А., Сенокосов А. Программно-методический комплекс для классов с углубленным изучением ОИВТ	21
Брусиловский П., Горская-Белова Т., Зырянов М. Интеллектуальная компьютерная среда для обучения физической географии	25
Многопользовательский персональный компьютер	30

Кабинет ВТ

Бакман Е., Дремова И. Введение в микро-Пролог	37
Ковтун М. Программирование в системе Express Pascal	41
Фуфыгин Я. Новый текстовый редактор для «Агата»	44
Ваграменко Я., Колыхалов П., Любавина Г. Банк профессиональных знаний	46
Луцкий В., Шахнович В. Графика «Агата» — новые возможности	48
Швалев В. Контроль знаний в дисплейном классе	50
Сочнев С. Пример контролирующей программы	53
Софронова Н. Использование графических возможностей «Синклер Спектрум»	57
Федорова М. Физиологическое обоснование режима обучения школьников при работе на ЭВМ	58

Клуб «Корвет»

О «Корвете»	61
Ахманов С., Нечаев А., Рой Н., Скурихин А. Архитектура «Корвета»	63
Ахманов С., Нечаев А., Скурихин А. Архитектура процессора КР580ВМ80А	66
«Корвет»-информ	69
DRAW на РМУ	71

Клуб БК

Текстовый редактор для БК: новые принципы и возможности	72
BIGRED — текстовый редактор для БК-0010(01)	75
Попробуйте так!	76
Панченков И. Подключение принтеров к БК-0010(01)	78
Операционная система для БК-0010	81
Программное исполнение системных команд Бейсика БК	82
Кирпичики ваших программ	83
Аляев Ю., Кузнецов С. О возможностях функции MID\$ при работе с файлами на КУВТ-86	85

Педагогический опыт

Голыцина И. Использование оболочки ЭС для создания ППС	88
Романов В. Новый вариант школьной программы	91
Вечкутова М., Дрягина М. Школьная электронная почта	94

Внеклассная работа

Цикоза В. Альтернативная задача II тура	98
---	----

Молодежная инициатива

105

Зарубежный опыт

Компьютеры в обучении: шведский путь	112
--------------------------------------	-----

Нам пишут

Учитель — менеджер	120
Автоматизированный учебный курс «Электрик» для КУВТ «Ямаха MSX-2»	121
Делим с любой точностью	122
Анкета социолога	123

Информация

Ассоциация учителей информатики — производителям ВТ, разработчикам программных средств	127
Методические рекомендации по созданию и использованию педагогических программных средств	127

Главный редактор
В. А. МЕЛЬНИКОВ

Редакционная
коллегия

И. Н. АНТИПОВ
В. Н. АФАНАСЬЕВ

И. М. БОБКО
Г. В. ГОДЖЕЛЛО

Б. В. ЛОМОВ
Ю. В. ЛУИЗО

(зам. главного
редактора)

Н. Г. МЕЛЬДИАНОВ
И. С. ОРЕШКОВ

О. К. ПАВЛОВА
А. Ю. УВАРОВ

А. И. ФУРСЕНКО
В. О. ХОРОШИЛОВ

К. В. ШЕХОВЦЕВ
(редактор отдела)

Обложка С. Плюща

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

Сдано в набор 25.11.91. Подписано в печать
Формат 70×100^{1/16}. Бумага офсетная. Печать офсетная.
Усл. печ. л. 10,40. Усл. кр.-отт. 42,88. Уч.-изд. л. 13,37
Тираж 55055 экз. Заказ 1856. Цена 2 руб.

Издательство «Педагогика» Академии педагогических наук СССР
Почту направлять по адресу: 119034, Москва, Смоленский б-р, д. 4. Издательство
«Педагогика». Журнал «Информатика и образование».
Адрес редакции: Лефортовский пер., д. 8.
Телефон: 261-11-29

Ордена Трудового Красного Знамени Чеховский полиграфический комбинат
Министерства печати и информации Российской Федерации
142300, г. Чехов, Московской обл.

© «Педагогика», «Информатика и образование», 1992

М. ЛАПЧИК

проректор по научной работе Омского педагогического института

Информатика и технология: компоненты педагогического образования

Из чего складывается информационная и технологическая грамотность современного учителя?

Знания в области информатики. Вспоминая те времена, когда курс ОИВТ вводился в школу (краткосрочные летние курсы при ИУУ, ускоренные «пожарные» меры в педвузах по корректировке учебных планов выпускных курсов, привлечение в школы людей с инженерными знаниями и т. п.), можно сказать, что учитель шел в класс практически безоружным. Идеолог первой программы курса ОИВТ академик А. П. Ершов в то непростое время призывал учителя откровенно говорить ученикам, что он сам ничего не знает и будет учиться вместе с ними. А речь-то шла фактически об овладении «азами» школьного Е-языка и Бейсика.

Теперь такое уже «не проходит», школе нужен учитель информатики с фундаментальными знаниями в области информатики. Причем объем этих знаний неожиданно (а для некоторых — и просто непредвиденно) не только стал уже достаточно осязаемым, но и имеет тенденции к постоянному (и довольно энергичному) возрастанию.

Это прежде всего хорошее знание ЭВМ, ее архитектуры. И здесь обоснование кроется уже не в традиционном объяснении, суть которого в том, что знание архитектуры дает сознательное понимание возможностей и принципа работы ЭВМ — такое объяснение годится лишь для среднего школьника. Учитель информатики должен глубоко знать

ЭВМ уже просто потому, чтобы не оказаться учеником у любознательного учащегося, который сегодня нередко с удовольствием программирует в кодах.

Далее. Широкая теоретическая и практическая эрудиция в области операционных систем и языков (систем) программирования: и здесь уже, разумеется не только (и не столько) пресловутый Бейсик, надолго застрявший в школьной программе курса ОИВТ; нужны знания самых разных мощных современных языков: Паскаль, Си, Пролог и др.

Работа с программным инструментарием становится сегодня неотъемлемой частью нормального образования учителя информатики (да и не только информатики!). Особенно актуальность этого раздела практического знания возросла в связи с достаточно уже широким распространением в школьной сфере IBM-компьютеров, оснащенных пакетом мощных инструментальных программных средств западных фирм, освоение которых составляет особый раздел приложений информатики. Это связано, в частности, и с задачей разработки ПС школьниками под руководством учителя — надо оберегать школьников от кустарных подходов к созданию программного продукта, показывать ему новые подходы. А для этого приходится волей-неволей переходить к новым эффективным технологиям создания программных средств с использованием баз данных и баз знаний, экспертных систем.

Умение работать в коммуникационной компьютерной среде — дело пока достаточно новое для нашей практики, нередко все ограничивается только теоретическими размышлениями на эту тему да

Окончание. Начало см.: Информатика и образование. 1991. № 6.

впечатляющими государственными программами, о которых говорилось выше. Но время идет, и очень скоро соответствующие представления и практические навыки (будем надеяться!) станут для нас столь же обычными, как навыки пользования телефонным аппаратом. Сейчас же надо побольше узнавать в этой области, готовиться к обыденному использованию таких понятий, как «модем», «хост-машина», «телеконференция», «электронная почта» и т. п. Пройдет совсем немного времени, и эти знания станут рядовыми для любого учителя, как и для любого члена нашего общества, когда все это войдет в разряд обыкновенной привычки пользоваться средствами новых коммуникационных технологий.

4 Знания в области педагогической технологии. Современные инновационные подходы круто меняют многие веками складывавшиеся представления о самой сути педагогического труда — сегодня это напрямую связывается и с аспектами информатизации. Много слов говорилось уже по поводу того, что такое урок в компьютерном классе, однако на деле эта новая область старой педагогики и методики преподавания, которую теперь чаще называют словом «технология», разработана чрезвычайно слабо. И тем не менее соответствующее образование в этой области учитель должен иметь. Что здесь?

Прежде всего — хорошее знание и владение функциональными возможностями локальной сети КВТ, а далее — как говорилось выше — и знание основ глобальных компьютерных систем связи. Кроме того, дидактика и методология школьного урока на базе комплексно оснащенного КВТ (локальная сеть ПЭВМ, видео и другая программно-управляемая аппаратура) с возможностями программирования и автоматизированного управления совокупностью обучающихся и педагогических воздействий. Пока что развитие этой новой отрасли педагогического образования с трудом вырывается из традиционных и незыблемых «оков» дидактики — таких, как урок, ученическая группа (класс) и т. п. Однако уже то, что сегодня приходится делать учителю в условиях преподава-

ния в типовом компьютерном кабинете, оснащенном пакетом программных средств: планировать урок с дозированием времени на обращение к ППС, соединять традиционные профессиональные приемы и возможности локальной сети КВТ, вводить новые формы индивидуального обучения и фронтального контроля знаний и т. п., — все это заставляет основательно пересматривать багаж профессиональной подготовки современного учителя.

Настойчивые требования школьной практики «подгоняют», не позволяют задерживаться с пересмотром и корректировкой программ подготовки не только учителей информатики, но и учителей всех других специальностей в педагогических учебных заведениях.

Обеспечивает ли действующая система подготовки учителей в педагогических учебных заведениях требуемый современный уровень подготовки молодого специалиста?

К сожалению, не обеспечивает.

Первые учебные планы физико-математических факультетов педвузов с достаточно развитым перечнем «компьютерных» дисциплин введены Минпросом СССР, как известно, в 1985 г. Основой планов послужил опыт Свердловского и Омского педвузов.

Был существенно расширен (по сравнению с предыдущими вариантами планов) курс ОИВТ — с 1-го по 5-й семестр; курс ТСО (НИТО). Введены две учебные практики: применение НИТ к решению задач предметной области (практика на ПЭВМ) и технологическая практика (практика в КВТ); информатизированные курсы методики преподавания предмета (математики, физики); педагогическая практика в школе (с проведением уроков в КВТ). Введение указанного цикла дисциплин в учебные планы «продавливалось» в условиях монологического протеста со стороны радетелей за спокойное положение и объемы прочих дисциплин. В педвузах, принявших эти учебные планы, накоплен определенный опыт, наработано немало нового учебно-методического обеспечения. Сегодня можно сказать, что основной замысел того поколения учебных планов был совершенно правильным.

Хотелось бы отметить ключевую роль двух учебных практик, выполняющих кроме всего прочего важную функцию мобилизации в повышении квалификации преподавательских кадров кафедр факультетов. Первая учебная практика (практика на ПЭВМ) нацелена на формирование у студентов навыков решения прикладных задач профильной предметной области (для физматов — математики, физики) с помощью ПЭВМ. К проведению практики привлекаются исключительно преподаватели специальных кафедр факультета. Форма организации практики может быть «плавающей»: за преподавателем закрепляется некоторое число (5—7) студентов, которым выдаются индивидуальные задания, предоставляется необходимое время для работы на ПЭВМ, преподаватель консультирует, контролирует и принимает работу студентов в пределах установленного деканатом времени. Для проведения второй учебной практики (практики в КВТ) привлекаются исключительно методисты факультетов (математики, физики, информатики). Практика обычно проводится по расписанию в КВТ с функционирующими локальными сетями с подгруппами в 10—12 человек. Цель практики — отработка навыков использования функциональных возможностей локальной сети ПЭВМ (а если имеются — и других технических возможностей: проекционной аппаратуры, видео и т. п.) для достижения дидактических целей урока. Под руководством преподавателя-методиста студенты учатся конструировать и практически апробировать уроки на базе КВТ, привлекая конкретные разделы школьных дисциплин. В этом отношении вторая учебная практика служит как бы концентрированным продолжением лабораторно-практической части по курсам методик преподавания. Наиболее целесообразное время для проведения этой учебной практики — до выхода студентов в школу на завершающую стадию педагогической практики. Как показывает опыт, широкое привлечение к проведению учебных практик практически всех преподавателей факультетских кафедр служит весьма полезным организующим фактором, создает своеобраз-

ную «компьютерную» атмосферу на факультете, резко меняет отношение к этому делу не только преподавателей, но и студентов.

Дисциплины цикла компьютерной и технологической подготовки	Семестры											
	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII
	1-я				2-я				3-я			
ИНФОРМАТИКА: основы информатики; работа с инструментарием НИТ в предм. области; применение в творческой работе	*	*	*						*	*		
НИТО: средства НИТО; дидактика и методология НИТО; применение в творческой работе				*				*			*	
МЕТОДИКА (И) ПРЕПОДАВАНИЯ ПРЕДМЕТА (ОВ): 1-й концентр; 2-й концентр				(*)	*	*			*	*		
УЧЕБНЫЕ ПРАКТИКИ: практика на ПЭВМ; практика в КВТ				*				*			*	

	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII
ПЕД. ПРАКТИ- КА: 1-й кон- центр; 2-й кон- центр; 3-й кон- центр						*	(*)					
										*	(*)	
											*	*

За прошедший период идейные начала учебных планов физико-математических факультетов педвузов в области компьютеризации получили дальнейшее развитие.

В настоящее время на повестке дня —

проработка вариантов учебных планов педвузов, обеспечивающих ступенчатую подготовку специалистов. Проект вложения компьютерных компонентов в трехступенчатый вариант учебного плана педвуза приведен в таблице.

Не опоздать с приведением учебных планов в соответствие с современными требованиями к информационной и технологической грамотности будущего учителя, прорваться через остаточный принцип формирования компьютерного цикла дисциплин в учебных планах педвузов — актуальная задача для методических органов педагогических учебных заведений и министерств образования.

6

В. БЕЛОШАПКА

Информатика как наука о буквах

В статье продолжается обсуждение общих положений, на которых основано наше понимание феномена информатизации (см. [16], [17]). Это обсуждение непосредственно связано с проблемой построения курса информационного моделирования (см. [18]), с проработкой структуры его базисных понятий. Работа еще не завершена, но мы полагаем, что такой «текущий отчет» может быть интересным для читателей журнала.

Живая научная деятельность, в известной степени, свободна по отношению к наперед заданной структуре идей и понятий. Сами идеи и понятия рождаются и видоизменяются в той умственной среде, которая формируется вокруг хороших задач и крупных научных направлений. Иначе обстоит дело со школой. Логика жанра неумолимо требует фиксации совокупности базисных понятий и структуры их отношений.

Содержание статьи может быть представлено в виде следующих тезисов:

1. Следует различать компьютер как груды железа, пластика и кремния и компьютер как понятие. Компьютер как груды железа не может быть включен ни в научный, ни в педагогический контекст, поэтому мы, говоря о компьютере, понимаем его как понятие.

2. Компьютер имеет дело только с текстами, т. е. с конечными последовательностями букв. Все прочие объекты, процессы, явления и пр. попадают в сферу влияния компьютера лишь после представления в виде текста,

т. е. после информационного моделирования в виде информационных моделей.

3. Буква (литера), в том смысле, как это понимается в информатике, характеризуется следующими свойствами:

самотождественность ($A=A$);

дискретность ($A \neq B$);

финитность ($A, \dots, Я$);

толерантность (отсутствие априорной привязки к фиксированной смысловой единице, отсутствие значения).

4. Следует различать знак как символ, т. е. органическое единство смыслового и внешне выразительного аспектов, и знак как букву со свойством толерантности (см. п. 3). Все древние знаковые системы — это символические системы. Символизм пронизывает древние культуры, религии, цивилизации. Человек в этой картине мира не властен над символом, напротив, символ владеет человеком.

5. В происхождении буквы можно выделить ряд периодов, но использование Франсуа Виетом (XVI в.) букв в качестве имен для числовых величин ознаменовало собой появление буквы в современном смысле этого слова и начало всеобщей информатизации.

6. Концепция информационного моделирования, в известном смысле, завершила формирование новой (Галилей, Декарт, Ф. Бэкон и др.) парадигмы познания, которая строилась в режиме полемики со старой, созер-

цательной концепцией, и ее отрицания. Причем позиции этой новой концепции были на последней фазе откорректированы. «Миром правят числа» → «Миром правят буквы»; «Все должно быть измерено» → «Все должно быть обозначено».

После появления буквы появление компьютера было делом техники.

7. История информатизации математики поучительна тем, что она позволяет увидеть границы метода. История информатизации математики — это история того, как буква не смогла победить бесконечность. Математика не живет в прокрустовом ложе финитных концепций. И формулируя общее положение, можно сказать так: объект недоступен точному информационному представлению настолько, насколько он причастен актуальной бесконечности.

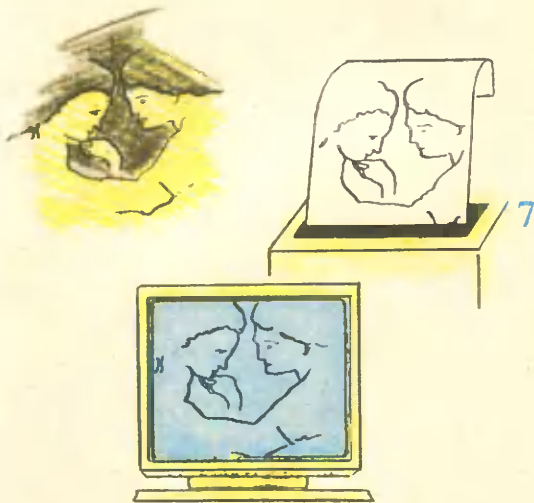
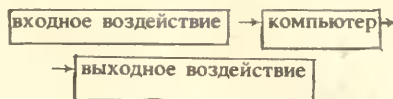
§ 1. Компьютер как понятие

Наше непосредственное восприятие мира — это чувственное восприятие. История человечества свидетельствует, что всякая зрелая духовная или культурная традиция, перерастая эту младенческую фазу, формулирует свой вотум недоверия непосредственно-чувственному восприятию как надежному инструменту познания. Не является в этом отношении исключением и европейская естественнонаучная традиция. «Ощущения — это обман наших чувств» (Декарт). В прекрасной книге американского математика Мориса Клайна [1] есть глава «Обман чувств и интуиция», где этот тезис подкрепляется целым спектром убедительных примеров, некоторые из которых широко известны. Итак, не останавливаясь на этом, мы подтверждаем наше стремление строить свое понимание на более прочном основании.

Однако положение вещей таково, что начальное знакомство с предметом — это чувственное восприятие. А чувства с неумолимостью свидетельствуют о том, что информатика — это компьютер. Мы не будем протестовать против такой постановки вопроса, но, не останавливаясь на этом, перейдем к обсуждению того, что такое компьютер. При этом мы надеемся, что внимательное рассмотрение, независимо от исходной точки, выведет нас на узловые понятия.

По-видимому, многим приходилось слышать, а некоторым даже читать в учебниках, что компьютер — это устройство для обработки информации; что информацией является почти все на свете, а может быть, даже вообще все; что информация бывает различных типов: графическая, звуковая, числовая и пр., что компьютер годится для обработки информации всех типов, а также для

преобразования одного типа в другой. В самом деле, что мешает нам взять полотно Рубенса, с помощью фотоввода записать его в память компьютера, подвергнуть какой-либо обработке и вывести на печать? Итак, схема такова:



Но если мы ограничимся этой схемой, то вскоре убедимся, что она имеет чрезвычайно общий характер. Схема одинаково хорошо описывает работу фотоаппарата, радиоприемника, зеркала, поэта и пр. и очень слабо отражает специфику компьютерных форм обработки воздействий. Так что же выделяет компьютер и компьютерные формы обработки чего-либо от прочих? В чем идея компьютера? Ответ хорошо известен. Все компьютерные системы при работе с любыми входными воздействиями прежде совершения операций по их обработке совершают нечто похожее на приведение к единому общему знаменателю — представляют их в виде конечной последовательности букв некоторого конечного алфавита.

Это и есть тот существенный принцип, который формирует понятие «компьютер» и критерий, который позволяет квалифицировать каждый конкретный случай. Например, система для цифровой звукозаписи — компьютерная система, а аналоговая вычислительная машина для решения дифференциальных уравнений — это нечто иное.

Различные технические реализации ис-

пользуют алфавиты с разным количеством букв. Возможность перекодировки, т. е. записи букв одного алфавита сочетаниями букв другого, делает все конечные алфавиты (существенно, что букв не меньше двух) равнозначными с точки зрения задач компьютерной обработки воздействий. Как известно сейчас, наибольшим распространением пользуется двухбуквенный алфавит. И так, компьютер работает с текстами, т. е. с конечными последовательностями букв конечного алфавита.

Для того чтобы совершить произвольное преобразование текста, т. е. заменить его на любой другой текст, достаточно иметь возможность совершить замену буквы в любой позиции на другую произвольную букву. Как известно, все современные компьютеры предоставляют такую возможность. Следует отметить, что способы организации набора букв в компьютерах (последовательности, многомерные массивы и пр.), а также конкретный набор операций для осуществления преобразований и способы организации выполнения этих операций могут сильно различаться (различия в архитектуре конкретных компьютеров). Но в результате десятилетий интенсивных исследований был сформулирован и признан верным тезис Черча (см. [2], с. 24), который утверждает, что все разумные способы определения конструктивных объектов и процедур работы с ними (вычислимости) эквивалентны, в частности (и это является точной теоремой), все известные подходы дают эквивалентные результаты. В качестве примеров приведем следующий список ([2], с. 26):

исчисление равенств — Эрбран и Гедель
 μ -рекурсивные функции — Клини
 λ -определимость — Черч
вычислительные машины — Тьюринг
канонические системы — Пост
нормальные алгоритмы — Марков

Подытожим наши рассуждения. Объект, явление, процесс и пр. любой природы становится доступным компьютерной обработке только после представления его в виде текста над конечным алфавитом. Такое представление объекта называется его информационной моделью. Компьютер — это универсальное устройство для работы с текстами.

И наконец, мы не разделяем той романтической точки зрения, в соответствии с которой благоуханье трав, брачный зов марала и последний луч заходящего солнца являются объектами информационной природы. Трава — это трава, зов — это зов, и становятся они информационными объектами лишь в наших модельных построениях после представления их в виде текста.

§ 2. Буква как фундаментальное понятие информатики

Ввиду того значения, которое имеет для нас буква, нам следует остановиться на этом понятии для его внимательного исследования.

1. О тленности и непостоянстве вещей этого природного мира сказано немало. Но буква, как, впрочем, и иные обитатели мира идей и понятий, не участвует в бурных событиях, сотрясающих поднебесную. Буква не зависит ни от времени, ни от места, ни от температуры окружающей среды. Буква демонстрирует завидную *самотождественность*:

$$A=A$$

Конечно, это относится к букве как к понятию, а не к ее реализациям в виде записей.

2. Две различные буквы различны настолько фундаментально, что ничто не может заставить букву А стать равной букве Б. Так же, впрочем, как число 2 стать равным числу 3. Другими словами, буква — это объект *дискретный*.

3. Буква всегда выступает в составе некоторого фиксированного конечного набора букв, называемого алфавитом. И те конструкции, для строительства которых используются буквы, т. е. тексты, неизменно конечны. Этим мир букв кардинально отличается от мира чисел. Это свойство буквы мы называем *финитностью*.

Таким образом, мы отметили три свойства, которые характеризуют букву как понятие: самотождественность, дискретность и финитность. Но здесь еще нет самого главного.

4. Буква, как она понимается в информатике, обладает следующим уникальным свойством:

Буква, априори, не связана ни с какой смысловой единицей.

Благодаря этому, она готова соединиться с любой такой единицей в акте обозначения, образуя тот комплекс, который называется величиной. В результате мы можем написать $A:=2$. Это феноменальное свойство буквы мы назовем *толерантностью*. Это свойство резко выводит букву из круга всех прочих идей и понятий, и оно же превращает алфавит в универсальную среду для построения моделей.

§ 3. История буквы

Попробуем теперь проследить тот ряд метаморфоз, которые претерпела буква для достижения ею своего сегодняшнего статуса [3].

Наиболее древние формы письменности, которые с трудом можно вычленишь из современных им форм наскальной живописи,



принято относить к позднему палеолиту или раннему неолиту (условное начало неолита — VIII тыс. до Р. Х.). Это пиктография. Такого рода сообщение представляет собой последовательность рисунков. Пиктография была распространена у индейцев Америки, жителей тропической Африки, аборигенов Австралии, малых народов Сибири вплоть до XX в. Первоначальные «иконические» формы пиктограмм обладали непосредственным сходством с изображаемыми объектами. На их основе сформировались «символические» пиктограммы, которые это сходство утратили. Пиктография, в отличие от позднейших форм письма, — это способ непосредственной передачи сообщения чисто изобразительными средствами, вне связи со звуковой речью.

Однако в более поздних развитых системах пиктографии, позволявших передавать сложные сообщения, такая связь эпизодически возникала в виде элементов «ребусного написания», т. е. использования знака не по его основному значению, а в переносном смысле, по звуковой аналогии, по сходству звучания при произношении. Этот прием широко используется в современных ребусах.

Этот элемент «ребусного написания», получая большее распространение, осуществляет переход к письму, фиксирующему связную звуковую речь (протошумерское и протоэламское письмо, кохау ронго-ронго о. Пасхи, древнейшая китайская иероглифика). Итак, древнейшие фонетические иероглифические системы письма произошли от пиктографии. Вопрос о наличии единого центра происхождения остается в лингвистике открытым. По мнению сторонников моноцентристской гипотезы, этим центром был Шумер. Приведем даты, начиная с которых данное письмо известно: древнее египетское письмо — конец IV тыс. до Р. Х., шумерское письмо — начало III тыс. до Р. Х. (основа клинописи и эламской иеро-

глифики), протоиндское письмо — начало III тыс. до Р. Х., критское письмо — начало II тыс. до Р. Х., китайское письмо — II тыс. до Р. Х., письмо майя — I тыс. до Р. Х.

Все эти системы относятся к типу словесно-слоговых. Они состояли из базисной системы идеограмм (изобразительный элемент, связанный со смысловым комплексом), логограмм (знаков, обозначающих слово), знаков, уточняющих конкретную смысловую привязку идеограммы (знаков-детерминатив) и способов использования идеограмм для обозначения последовательности звуков путем их «ребусного» употребления.

В дальнейшем возникают системы письма, где знаки используются не для обозначения слов, а только последовательности звуков. Такие системы называются **силлабическими**. Силлабические системы — часто результат упрощения словесно-слоговых. Например, из критского письма, путем опущения логограмм, возникает кипрское письмо (6—4 вв. до Р. Х.). К этому же типу относятся индийское (3 в. до Р. Х.) и западно-семитское (первая половина — середина II тыс. до Р. Х.) письмо, которое и явилось историческим родоначальником всех видов алфавитного письма. Это письмо имеет структуру: (согласная+гласная) или просто (согласная) и происходит, вероятно, из финикийского протобиблейского силлабического письма с набором знаков около сотни.

В «классическом» финикийском алфавите (середина II тыс. до Р. Х.) имелось 22 знака. И, в отличие от силлабического письма, знак был предназначен для обозначения, по определенным правилам, элементарных звуковых единиц — фонем. В течение тысячелетия с ним успешно конкурировали словесно-слоговые системы письма. За это время он был воспринят в Малой Азии, Греции и Италии и дал начало всем известным алфавитным системам письма: западным — через греческий и восточным — через арамейский алфавит.

Здесь следует отметить, что все эти длительные и сложные метаморфозы не затронули той особенности, которая резко отделяет старое понимание слова «буква» от нового, в частности от того, которое используется в информатике. Речь идет о том, что в современной терминологии можно выразить так: знак и обозначаемое представлял собой органическое единство. И правильнее было бы говорить о единой сущности, у которой мы условно, с целью анализа, отмечаем эти два аспекта.

Причем не надо думать, что речь идет лишь о письменности. Мы полагаем, что здесь имеет место детерминанта того древнего строя сознания, который столь сильно отли-

чен от современного. Символы, т. е. именно такие органически единые комплексы, имеющие знаковое выражение, пронизывают все существование человека, — его культуру, религию. И повсюду мы встречаем именно страдательно-пассивное отношение индивидуального человеческого сознания к символу. Символы, и прежде всего письменные знаки, носили священный и сакральный характер. Во многих древних обществах грамотность была прерогативой жреческого сословия (иероглиф буквально — это священное изображение). Вся языческая теургическая практика, как и позднейшая оккультная, была тесно связана, если не сказать основана, на применении соответствующей символики. И вся она носила, по отношению к человеку, принудительно-обязательный характер. Причем это состояние несвободы пронизывало все аспекты жизни общества вплоть до хозяйственной жизни и ремесел. Так, например, для выплавки металла требовались человеческие жертвоприношения.

Не менее принудительным характером обладали буквы Ветхозаветного закона по отношению к древнему Израилю. «Проклят всякий человек, кто не исполнит всех слов закона сего и не будет поступать по ним!» (Второзаконие, гл. 27, ст. 26).

Это разительным образом отличается от того отношения к букве, к которому мы привыкли в контексте информатики. Когда, как и почему произошла эта перемена? Здесь вопрос. Ответ на него связан с глубинной мировоззренческой позицией. Для человека, который исповедует веру в прогресс и творческие возможности человеческого разума как самодостаточной сущности, это, по-видимому, очередная победа науки над мраком суеверий и предрассудков. Но мы смотрим на это иначе.

О значении пришествия Христа для самых удаленных от духовной проблематики культурных пластов и для индивидуального человеческого сознания, даже независимо от вероисповедания, сказано немало (см. например, [11]). Здесь мы позволим себе отметить лишь то, что имеет непосредственное отношение к нашему изложению.

Это, во-первых, необычный, по сравнению с тогдашней (маги, колдуны, волхвы, шаманы и пр.) и сегодняшней (йоги, биоэнергетики, экстрасенсы и пр.) оккультной практикой, характер Христовых и, вообще, новозаветных исцелений и чудотворений. Если для оккультного воздействия необходимы те или иные формы расслабления и отключения личной воли, то здесь ведущим моментом является свободный выбор человека: «Веруете ли в то, что Я могу это сделать? ... по вере

вашей да будет вам» (Евангелие от Матфея, гл. 9, ст. 28—29). Житийная литература полна свидетельств того, что над христианами утратила власть языческая магия.

А во-вторых, это совершенно иное, по сравнению с ветхозаветным, евангельское отношение к букве закона. За нарушение этой буквы Христа и распяли. Воскресение Христово — это освобождение от власти смерти и буквы. «И познаете Истину и Истина сделает вас свободными» (Евангелие от Иоанна, гл. 8, ст. 32); «Буква убивает, а дух животворит» (2-е послание к Коринфянам, гл. 3, ст. 6); «И уже не теснится в Законе человечество, но в Благодати свободно ходит» (Святитель Иларион, Слово о Законе и Благодати [5]).

Теперь мы можем завершить наше изложение истории буквы. Честь оформления этой, неведомой древнему миру, свободы обращения с буквой в виде рабочего научного понятия принадлежит, по-видимому, Франсуа Виету (1540—1603) [6]. Виет занимался тем разделом математики, который сейчас называется элементарной алгеброй. И он был первым, кто использовал буквенные обозначения для коэффициентов уравнений. До Виета никто не решался написать $A=2$. (Как известно, A , т. е. «Аз» — это 1, а 2 — это B , т. е. «Веди», а «Аз» и «Веди» путать не следует.)

На этом история рождения буквы заканчивается, и начинается история ее жизни, т. е. история информатизации.

§ 4. Новое пифагорейство, или история девальвации числа

Пифагору приписывают высказывание: «Миром правят числа». И эта формулировка является концентрированным выражением позиции всего античного символизма — традиции с тысячелетней историей от Пифагора через Платона до Прокла [7]. Современные исследователи отмечают парадоксальный временной разрыв между периодом существования этой традиции и периодом блестящего подтверждения данного тезиса в успехах математического естествознания Нового времени [1]. Действительно, тот букет достижений в области изучения природы ясно свидетельствовал о том, что законы природы — это числовые соотношения.

В это время в работах Декарта (1596—1650), Галилея (1564—1642), Ф. Бэкона (1561—1626), Ньютона (1643—1737), Гюйгенса (1629—1695), Лейбница (1646—1716) формируются основы современных представлений о том, что такое хорошо, правильно построенное научное исследование. В этих представлениях центральные позиции принадлежали числу, измерениям и количе-

ственным соотношениям. При этом вера в то, что законы природы написаны на языке математических формул, предшествует и эксперименту, и открытию. Вот фрагмент из «Диалога о двух главнейших системах мира» Галилея [8]:

Симплицио: «Как же это, не проделав ни ста испытаний, ни даже одного, вы выступаете столь решительным образом?»

Сальвиати (выражающий взгляды Галилея): «Я и без опыта уверен, что результат будет такой, какой я вам говорю...»

И это, несомненно, уверенность, унаследованная от античности.

Однако у того же Галилея звучит новый, совершенно чуждый античности, мотив, суть которого в том, что «положительное физическое знание следует отделять от вопросов о причинной зависимости, а всякого рода предположения о физических причинах оставить в стороне. Галилей настоятельно советовал естествоиспытателям: «Не рассуждайте о том, почему происходит какое-то явление — описывайте его количественно» [1], с. 112). Такая позиция Галилея вызвала протест даже у Декарта: «Все, что Галилей говорит о телах, падающих в пустоте, лишено всякого основания; ему следовало бы сначала определить природу тяготения» (там же). Но именно она, эта позиция, стала программой для новой натурфилософии от Галилея до наших дней: «...оставим в стороне вопрос о том, почему Природа устроена так, а не иначе; для объяснения этого хороших теорий нет» (Фейнман, [9], с. 15).

В соответствии с этим возникает важное отличие пифагорейско-платоновского понимания числа от его понимания у исследователей Нового времени. Если для античного символизма «число является не чем иным, как структурой всей действительности в целом» ([7], т. 7, кн. 1, с. 155), то европейская научная традиция игнорирует в понятии «число» весь объем его содержания, сохраняя лишь момент чисто количественный. При этом напомним, что изобретение Виета (см. § 3) было уже в научном обиходе и девальвация смысла понятия «число» происходила параллельно наращиванию его информационного компонента. Это создавало возможность строить процедуры работы с числовыми величинами без апелляции к смысловой составляющей, а лишь на основе работы с обозначениями, с буквами.

И недаром тогда же прозвучал первый призыв ко всеобщей информатизации. Он принадлежал Лейбницу. Лейбниц предлагал создать некое универсальное буквенное исчисление, позволяющее правильно отражать исходные предпосылки и законы мышления, после чего «все научные, политические,

нравственные и философские проблемы могли бы разрешаться путем механических вычислений так же, как, например, выполняется умножение двух целых чисел. Драматическая история информатизации математики закончилась в определенном смысле в 30-х гг. XX в. доказательством невозможности реализации предложения Лейбница даже, в рамках математики (теоремы К. Геделя, см. [10], где дано доступное и увлекательное изложение).

Три столетия триумфального развития европейской науки (XVII, XVIII и XIX вв.) не изменили основных представлений о научности и задачах научного исследования. Эпицентром движения оставалось математическое естествознание, а его ведущим методом — метод математического моделирования. Но в конце XIX в. и начале XX в. произошли бурные события, превратившие информатизацию в общенаучное движение [16] с информационным моделированием в качестве ведущего метода. Это произошло за счет следующей и, по-видимому, последней девальвации числа: был ампутирован рудимент смысловой компоненты этого понятия (количество) и число превратилось в ту букву, о которой шла речь выше. В подготовке этих событий не последняя роль принадлежит И. Канту (1724—1804), который, не отрицая существования у вещей глубинно-онтологических пластов («вещь в себе»), утверждал, что они для нас недоступны, а также Ч. Пирсу (1839—1914) и Г. Фреге (1848—1925). Лишившись остатков смысловой нагрузки, число, превратившись в букву, стало доступно автоматической обработке с помощью специально созданного технического устройства — компьютера.

Триумфальное шествие буквы, под знаком которого находится наука XX в., связано с определенным мировоззрением в среде исследователей. Его мы квалифицируем как новое пифагорейство. Это позиция, в которой пифагорейский принцип «миром правят числа» получает реакцию «миром правят буквы», а галилейский тезис «все должно быть измерено» переходит в основной тезис информационного моделирования «все должно быть обозначено». Противопоставление старого и нового пифагорейства — это противопоставление двух концепций познания. Древней, созерцательно-медитативной, в том или ином виде присутствующей во всех имеющих солидные исторические корни духовных и культурных традициях, которая понимает познание как событие, т. е. совместное бытие познающего субъекта и познаваемого объекта, утверждая ведущую роль онтологии по отношению к гносеоло-

гии. И новой, вызревшей за последние четыре века в рамках европейской научной традиции, основанной на объект-субъектном противопоставлении, тяготеющей к чисто гносеологическим постановкам и избегающей онтологической проблематики.

В этой новой концепции познания нет места для вопросов, находившихся традиционно в центре внимания: в чем смысл человеческого существования? Чем жив человек? Почему в мире столько страдания и зла?

Они объявляются плохо поставленными вопросами, порождением неразвитых языковых форм, а размышление над ними — занятием бесполезным. Да и сам человек, а тем более информатизированный человек, член информатизированного общества, теряет свой прежний статус, интегрируясь технологической средой. Об этом, на наш взгляд, хорошо написано у Бердяева [11] и у Гвардини [12]. Та свобода от буквы, которую принес человечеству Сын Божий, на наших глазах ради технологической и гносеологической эффективности утрачивается и возвращается букве в ее новом, компьютерном исполнении. У Вейценбаума ([13], с. 303) анализу этого феномена уделяется серьезное внимание. Там, в частности, обсуждается история времен вьетнамской войны, когда компьютерная система, оказавшись фактически бесконтрольной, самостоятельно принимала решения о назначении бомбардировок по вьетнамской территории. В качестве еще одного симптома можно указать на рост интереса к астрологии, концепция которой содержит представления о власти буквы (астрологического знака) над человеческой судьбой и иллюзорности свободы человеческого поступка. Эта проблема, на наш взгляд, стоит для человечества остро. И здесь главную опасность представляют не возможные политические или социальные бедствия, хотя они возможны, а именно опасность порабощения сознания, подобная той, что владела человечеством в язычестве. Ф. Бэкон говорил о порабощении сознания идолами и даже перечислял их. Современное сознание попадает в порабощение идолу в виде отлитой в буквах и овеществленной в компьютере модели.

Но вселяет оптимизм история информатизации математики, которую можно понимать, как историю того, как буква не смогла

победить бесконечность. Математике было, в некоторый момент, предложено отказаться ради эффективности ее построений от любимой ею актуальной бесконечности (см. [14], с. 493 и [15]), но математика не смогла с ней расстаться и поглощение ее информатикой не состоялось. Мы полагаем, что судьба человеческого сознания, и вообще, и личного, в плане полной и окончательной информатизации, зависит от его непосредственно-жизненной причастности Сущности, существенно и актуально бесконечной.

Было бы в самом деле нелепо и печально, если бы человеческое сознание пало жертвой им же созданного инструмента. Итак, осталось еще раз повторить слова Спасителя: «И познаете Истину и Истина делает вас свободными».

Литература

1. *Клайн М.* Математика. Поиск истины. М.: Мир, 1988.
2. *Мартин-Леф П.* Очерки по конструктивной математике. М.: Мир, 1975.
3. Лингвистический энциклопедический словарь. М.: Советская энциклопедия, 1990.
4. *Миняя, Октябрь.* Житийные справки (приложение). Издание Московской Патриархии, 1980.
5. *Миняя, Июль.* Ч. 3. Издание Московской Патриархии, 1988.
6. *Стрѣйк Д. Я.* Краткий очерк истории математики. М.: Наука, 1984.
7. *Лосев А. Ф.* История античной эстетики, т. 7, книга 1. М.: Искусство, 1988.
8. *Галилей Г.* Избранные труды в двух томах. М.: Наука, 1964.
9. *Фейнман Р.* КЭД — странная теория света и вещества. М.: Наука, 1988.
10. *Смаллиан Р.* Как же называется эта книга? М.: Мир, 1981; Принцесса или тигр? М.: Мир, 1985.
11. *Бердяев Н.* Смысл истории. М.: Мысль, 1990.
12. *Гвардини Р.* Конец нового времени // Вопросы философии. № 4. 1990. С. 127—163.
13. *Вейценбаум Дж.* Возможности вычислительных машин и человеческий разум. М.: Радио и связь, 1982.
14. *Флоренский П.* Столп и утверждение Истины. М.: Правда. 1990.
15. Математическая энциклопедия. Т. 1. М.: Советская энциклопедия, 1977.
16. *Белошапка В. К.* О языках, моделях и информатике // Информатика и образование. 1987. № 6.
17. *Белошапка В. К.* Три аспекта мироздания, или Мир как информационная структура // Информатика и образование. 1988. № 5.
18. *Белошапка В. К., Лесневский А. С.* Основы информационного моделирования // Информатика и образование. 1989. № 3.

АСУ обучением с минимальными потерями

Введение. Хорошо известно, что «обучение есть процесс передачи и усвоения знаний, умений, навыков деятельности...» [1, 136]. Из этого определения следует, что процесс обучения состоит из двух фаз:

фазы передачи знаний, умений, навыков;
фазы контроля усвоения.

Эти две фазы в современных системах компьютерного обучения связываются в систему с обратной связью посредством принципа упражняемости, или повторения. На входе такой системы знания, подлежащие усвоению; на выходе — усвоенные знания.

Но где гарантии, что усвоено всё? Где, например, гарантии того, что в памяти обучаемого сформировалась информационная структура, адекватная изучаемому материалу, что не произошло потери передаваемой информации? Насколько эффективна система в отношении потерь времени и усилий? Как объяснить и предупредить случаи неэффективного повторения или, другими словами, как организовать процедуру повторения?

Задача настоящей статьи — ответить на эти вопросы. Все они связаны с проблемой контроля усвоения, которая, в свою очередь, связана с проблемой функционирования психики человека, и в частности с функционированием рецепторной системы — системы восприятия и запоминания информации.

Последние исследования еще раз показали, что психика человека представляет собой регуляторное образование [2]. Это, в частности, справедливо и для памяти, значение которой для процесса обучения можно определить как решающее. В этой связи необходимым является рассмотрение вопроса регуляции и управления и в общем, и особенно в контексте процесса обучения, а также вопроса функционирования информационных систем организма, в частности памяти.

Обучение и управление. Под управлением в кибернетике понимают такое воздействие на объект (процесс), которое выбрано из множества воздействий с учетом поставленной цели, состояния объекта (процесса), его характеристик и ведет к улучшению функционирования или развития данного объекта (процесса) [3].

Комментируя это определение, Н. Ф. Талызина подчеркивает, что управлять — не значит подавлять, навязывать процессу ход, не соответствующий его природе, а наобо-

рот — максимально учитывать природу процесса, согласовывать каждое воздействие на процесс с его логикой. При управлении процессом учения, как и любым другим, свобода выступает как познанная необходимость [4].

Такое понимание управления адекватно определению, данному В. В. Смольяниновым: «Управление — целевая акция редукции избыточных свобод системной (структурной и/или функциональной) организации» [5, с. 107], которое укладывается в приведенную им же семантическую формулу для организации:

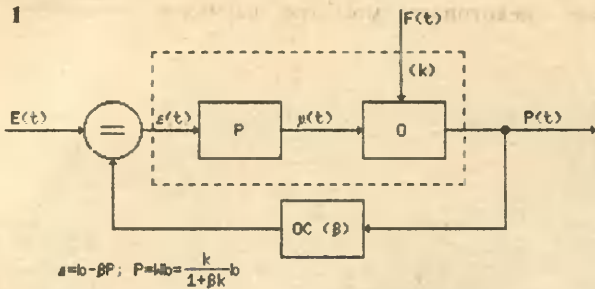
{избыточность, инварианты} → организация, что адекватно, по его словам, его определению управления, а значит, может быть представлено как

{избыточность, инварианты} → управление (1).

Смысл последней формулы должен пониматься так: *управление есть редукция (устранение) избыточности посредством выявления инвариантов с учетом поставленной цели.*

Для нас в приведенной формуле важно то, что В. В. Смольянинов отмечает в тексте своей статьи как принципиальный момент — вопрос о сущности управления. «Конкретные системы, — пишет он, — могут воплощаться на разных материальных носителях — механических, электрических и др. Тогда форма воплощения определяет природу управляющих переменных, но как раз материальное разнообразие воплощений — одно из свидетельств нематериальной сущности управления... сущность управления не просто в количественных различиях величин носителей (которые всегда материальны), а в том, что для управления существенны значения в символической значимости, а именно как нематериальные семантические сущности... — алгоритмы и информация...» [5, с. 107].

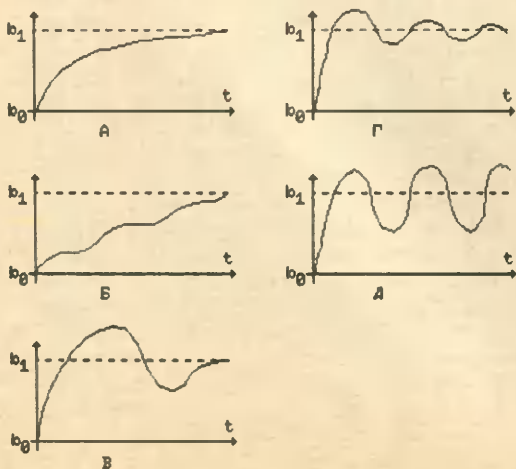
С учетом сказанного выше, т. е. рассматривая (1) с феноменологического уровня, под избыточностью степеней свободы можно понимать избыточность информации вообще, а под инвариантами — ценную информацию, которую выделяет управляющая система в соответствии с целями (законом) управления, чтобы использовать ее затем для осуществления целесообразных управляющих воздействий. Таким образом, *управление, а также обучение можно понимать как редукцию (устранение) избыточности ин-*



формации в соответствии с целью управления.

Все виды управления, как известно, можно подразделить на разомкнутое и циклическое управление. Первое — управление без обратной связи и, следовательно, без регуляции хода управляемого процесса со стороны управляющей системы. Второе предполагает наличие и того, и другого. Это более эффективный способ управления и наиболее характерный для кибернетики, поскольку он обеспечивает устойчивость и развитие системы. На рис. 1 представлена структурная схема управления с обратной связью. Символом К на нем обозначена некоторая характеристика (комплексный коэффициент усиления) системы, формирующей выходное воздействие $p(t)$. Сама система управляется воздействием $e(t)$, называемым сигналом рассогласования и представляющим собой некоторую функцию от выходного воздействия $p(t)$ и входного воздействия $b(t)$, подаваемых на сравнивающий элемент по каналу обратной связи. Символом $\beta(t)$ обозначена характеристика цепи обратной связи (коэффициент усиления); $F(t)$ — посторонние, неучитываемые возмущения, воздействующие на систему. Легко видеть, что подобную систему можно представить

2

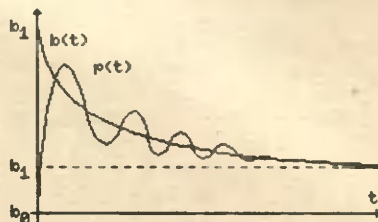


с более сложной, чем К, характеристикой W (см. рис. 1).

Изменение характеристики обратной связи ведет к существенному изменению поведения системы. На рис. 2 приведены характерные типы такого поведения при резком изменении входного воздействия от значения b_0 до значения b_1 . Поведение может быть монотонным, апериодическим, с перерегулированием, колебательным (рис. 2, б—д соответственно). В некоторых случаях при положительной обратной связи на выходе может возникнуть незатухающий колебательный процесс (рис. 2, е) [6].

В данном случае законом управления для управляющей системы является регуляция по значению $b(t)$. Но $b(t)$ может быть любой функцией, и в частности монотонно убывающей. И тогда поведение системы может иметь вид, показанный на рис. 3.

3



Управление всегда целенаправленно. В одних случаях целью управления может быть поддержание параметров объекта (процесса) управления О (см. рис. 1) в определенных пределах. В других случаях цель управления — отработать закон (программу) управления $b(t)$, т. е. перевести объект (процесс) управления О из одного состояния (с одним значением параметра b_1) в другое (со значением b_2), как это изображено на рис. 3. Другими словами, в общем случае важны значения как исходного состояния объекта управления, так и конечного. Информация о состояниях объекта (процесса) управления О, определяемая по его выходу, поступает по каналу обратной связи на сравнивающий элемент, где осуществляется выделение ценной информации, в соответствии с целью (законом) управления. На основании этой ценной информации в регуляторе Р вырабатываются сигналы коррекции — управляющие воздействия $\mu(t)$ на объект (процесс) управления О.

Может случиться, что внешние условия, которые определяют возмущающее воздействие $F(t)$, будут меняться столь стремительно, что достижение системой цели ста-

нет невозможным, не говоря уже о случае, когда такое воздействие окажется просто разрушительным для системы. В таких случаях принято говорить, что система не справляется с управлением или что управление является неэффективным. Для преодоления такой ситуации, для того чтобы управление было эффективным, для так называемой «свободы маневра» системе «необходимо создавать те свободы, те избыточные возможности, которые затем преодолеваются в акциях управления или самоорганизации» [5, с. 109].

Здесь как раз и проявляется диалектичность процессов управления. Для его успешного осуществления из процедуры редукции (устранения избыточности) информации самоорганизующейся, саморазвивающейся системе в общем случае необходимо осуществлять сбор, поиск самой разнообразной информации, создавать ее избыточность. Другими словами, для последующего выделения инвариантов системе необходимо наличие разнообразной информации, относительно которой в акциях управления и определяется свойство неизменности, т. е. инвариантности.

В этом и заключается фундаментальный закон: *управление является диалектическим единством процесса создания свобод (избыточности информации) и процесса их редукции (устранения избыточности) в соответствии с целью управления.*

Функции информационных систем живых организмов. Как мы уже отмечали, вопросы контроля усвоения информации в процессе обучения упираются в проблему восприятия информации человеком, т. е. в функционирование рецепторной системы. В психологии уже давно и прочно обосновалось представление, что психика представляет собой регуляторное образование. Как указывал еще И. М. Сеченов, «всякое целесообразное действие регулируется чувствованиями», и для его управления требуются сложные процедуры «обратной афферентации» (П. К. Анохин), «сенсорной коррекции» (Н. А. Бернштейн). Это означает, что с точки зрения теории управления эта проблема может быть обозначена как проблема функционирования информационных систем живых организмов [7].

Функционирование информационных систем живых организмов, по утверждению В. П. Зинченко [7], определяется прежде всего тем, что субъект живет в своем окружении и для своего сохранения потребляет из среды энергию и информацию. Потребление информации субъектом связано с необходимостью создания в его памяти перцептивной модели, которая с

некоторого момента начинает управлять его поведением. Для информационной системы субъекта, в памяти которого строится эта перцептивная модель, характерно то, что она в процессе построения непрерывно соотносит модель с оригиналом. Модель должна быть предметной и адекватно отражать условия стоящей перед субъектом задачи.

Поэтому точность опознания объектов и ситуаций представляет собой один из важнейших признаков, характеризующих информационную систему. Вторым таким признаком является время построения модели и время опознания объектов и ситуаций, совершающегося на основе уже построенных моделей.

«Вначале построение модели требует специальных действий и представляет для организма специальную задачу. По мере построения эта модель постепенно приобретает функции регулятора поведения. При дальнейшем обучении и тренировке эта модель выполняет функцию регуляции исполнительных действий все более и более совершенно. Этот процесс психологами издавна описывался как процесс автоматизации действий и навыков, а выработанные и заученные способы поведения назывались машинообразными действиями или автоматизмами. Такого рода автоматизация наблюдается не только в сфере практических действий, но также в сфере перцептивных, мнемических и интеллектуальных процессов...»

...Переход от системы развернутых действий по построению концептуальных моделей к непосредственно исполнительным действиям, в осуществлении которых видимым образом отсутствует деятельность по построению моделей, представляет собой одно из средств повышения оперативности психических процессов» [7, с. 185].

Точность и время опознания объектов и ситуаций вполне могут служить критериями эффективности управления поведением субъекта со стороны построенных им моделей. Но эта эффективность может служить также свидетельством обученности субъекта, а поэтому точность опознания и время опознания объектов и ситуаций одновременно могут рассматриваться как критерии обученности — приобретения определенного количества знаний, умений, навыков.

Эти критерии, однако, не могут рассматриваться как критерии усвоения, поскольку ни точность построения перцептивной модели, ни время опознания объектов и ситуаций не могут быть определены и измерены существующими методами и

средствами. Следовательно, в качестве критериев усвоения должны использоваться критерии воспроизведения материала, подлежащего усвоению.

Критерии усвоения. Исключительно плодотворные, на наш взгляд, исследования процесса воспроизведения материала, подлежащего запоминанию, были проделаны В. Я. Ляудис [8], которая обосновала и углубила положение о том, что запоминание связано с построением образа объекта, соответствующего задаче воспроизведения, а не является актом закрепления воздействующего объекта. Функция же мнемического действия, выполняемого в процессе запоминания, состоит в создании сознательного плана собственных процедур, их отборе в соответствии с будущим, т. е. в организации актуально наличного по образу будущего; стало быть, мнемическое действие является саморегулирующей системой.

16

Но до этого, как показал в свое время П. И. Зинченко, должно быть в достаточной степени развито познавательное, умственное действие, с тем чтобы оно могло составить ориентировочную основу формирования образа объекта в процессе регулируемого запоминания. Известны исследования Л. М. Житниковой, в которых было подробно прослежено изменение продуктивности запоминания по мере овладения классификацией как умственным действием. Опыты показали, что до начала освоения классификации обучаемые не принимают и не выполняют задачу произвольного запоминания. Это свидетельствует, что в произвольном запоминании при построении образа действительно имеет место саморегуляция. *Образ объекта может быть построен и без этой саморегуляции в результате действия классификации объектов.*

Таким образом, именно задача предстоящего воспроизведения определяет продуктивность процесса регуляции при построении образа, свидетельством чего и является эффект закрепления образа в памяти. Продуктивность же процессов регуляции проявляется *в степени готовности памяти к воспроизведению.*

Все сказанное выше еще раз подтверждает вывод, что *точность и время* воспроизведения как раз и являются критериями регулируемости и продуктивности акта запоминания, а стало быть, могут успешно использоваться в качестве критериев усвоения.

Однако в столь, так сказать, «голом» виде они могут играть лишь формальную функцию. Поэтому В. Я. Ляудис предложила в качестве критерия готовности памяти к

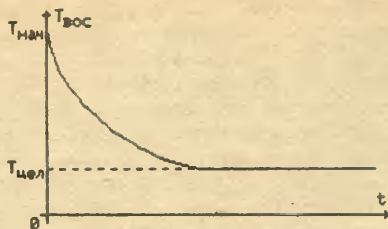
воспроизведению (а значит, и критерия усвоения) использовать *избирательность как точность и скорость воспроизведения по вариативным признакам*, поскольку важнейшим требованием, которое предъявляется к сформированным знаниям, является требование использовать материал в различных условиях и ситуациях (см. воздействие $F(t)$ на рис. 1). Тогда критерий избирательности как индикатор освоенности мнемического действия становится индикатором симультанного построения образа как целостной информационной структуры из отдельных компонентов. Это выражается в акте немедленного воспроизведения, которое нейтрализует роль отсрочки, совмещая подготовку к воспроизведению с моментом выполнения субъектом мыслительного действия в предъявленном материале, что становится свидетельством прочного усвоения знаний.

Автоматизированная система управления обучением. Изложенные выше положения общей теории управления и психологической теории восприятия и памяти позволяют по-новому взглянуть на перспективы развития машинного обучения.

С усвоением умственного и мнемического действия повышается точность воспроизведения образа предмета, предъявленного для запоминания, и уменьшается время воспроизведения этого образа; кроме того, существует очевидная связь между этими критериями, отражающими уровень готовности памяти к воспроизведению, проявляющийся в том, что чрезмерные ограничения на время воспроизведения приводят к потере точности воспроизведения. Следовательно, по определению управления, закон управления этим процессом должен быть адекватен процессу, т. е. он должен не подавлять и навязывать процессу ход, противоречащий его природе, а, наоборот, максимально учитывать природу этого процесса, согласовывать каждое воздействие на процесс с его логикой.

Этот закон управления должен осуществляться таким образом, чтобы управляющая система по каждой отдельной операции формирования умственного действия (см. [4] о пооперационном контроле) могла привести этот объект от некоторого уровня готовности памяти к более высокому уровню готовности, определяемому учебными требованиями и являющемуся целью управления.

Очевидно, что для времени воспроизведения ($T_{\text{вос}}$) график такого закона управления будет представлять собой монотонно убывающую экспоненциальную функцию, подобную представленной на рис. 4, где



$T_{нач}$ — начальное время воспроизведения, $T_{цел}$ — целевое время воспроизведения. АСУ обучением, предназначенная отслеживать такой закон управления, будет системой, управляющей скоростью обмена информацией, причем управляющей так, что эта скорость не будет выходить из диапазона скоростей бесстрессовой работы. Обучаемому, работающему с такой системой, всегда должно казаться, что скоростью обмена управляет он сам.

Такие системы известны как самонастраивающиеся, или адаптивные, обучающие машины. Впервые их предложил в 1954 г. известный английский ученый Гордон Паск [9].

По словам Г. Паска, в систему заложена способность к максимизации темпов обучения, что связано с так называемой операцией упрощения. «Предположим,— пишет Г. Паск,— что скорость увеличивается бесконечно (минимальное значение операции упрощения). Тогда материал станет непонятным. ...И наоборот, если скорость сокращена (максимальное упрощение), нет места никакому усовершенствованию и учащийся скучает. Таким образом, самонастраивающаяся обучающая машина должна сократить степень упрощения проблемы до предела, при котором показатель скорости обучения максимизирован. В этих условиях учащийся загружен соответственно своим способностям и его внимание поддерживается на должном уровне» [9, с. 403].

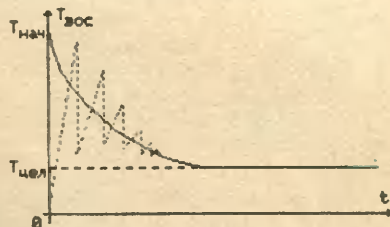
Хотя Г. Паск и утверждает, что обучение он рассматривает как управление учебным процессом, однако у него отсутствует четко сформулированная цель такого управления, а критерий управления (у него «темп обучения») он непосредственно не связывает с психофизиологическими состояниями, соответствующими уровням готовности памяти к воспроизведению. К тому же система, предложенная Г. Паском, ограничивалась использованием в системах, формирующих навыки и умения (тренажерах), в частности в системе, формирующей навыки быстрого чтения.

По нашему мнению, перенос акцента при рассмотрении критериев управления с «темпа обучения» на «точность и быстроту

воспроизведения» позволяет непосредственно связать критерии управления с психофизиологическими состояниями, соответствующими уровням готовности памяти к воспроизведению. Более того, такой подход позволяет связать возможность управлять не только с формированием навыков и умений, но и с передачей и усвоением информации, знаний. И в этом смысле знания не должны противопоставляться умениям и навыкам: знать означает выполнять какую-то деятельность или действия, связанные с данными знаниями [4].

Сам факт формулирования критериев и цели управления для управляющей системы является актом выделения инвариантов, посредством которых в процессе управления осуществляется редукция избыточности, что соответствует определению управления и согласуется с формулой (1). Справедливость этого очевидна. Например, график отслеживаемого обучающей системой времени воспроизведения для одной операции в зависимости от времени будет иметь вид (рис. 5), аналогичный графику возможного поведения системы управления (см. рис. 3).

17



5

Таким образом, время воспроизведения для одной операции формирования умственного действия, определяющее время взаимодействия обучаемого с системой, т. е. интервал взаимодействия (см. также [2]), как раз и будет инвариантом для обучаемого и обучающей системы, который отслеживается (редуцируется) системой, что со временем приведет к минимально возможной (эталонной) величине этого интервала. Подобным же образом избыточность возможных ошибочных ответов редуцируется постепенно к находящемуся в системе эталонному оригиналу, что обеспечивает точность воспроизведения, а в совокупности по двум критериям приводит к минимизации потерь времени и информации, приходящихся на одну операцию формирования умственного действия. Минимальность же временных затрат на передачу всей информации для одной операции формирования умственного действия определяет минимальность затрат энергии на одну операцию за весь период работы.

Возможность установки на пути к целевому состоянию — целевому уровню готовности памяти — промежуточных уровней, а также возможность пооперационного контроля усвоения позволяют осуществить селекцию уже сформировавшихся операций и не возобновлять операции передачи (коррекции), когда такой уровень достигнут, т. е. не осуществлять излишнее повторение, что должно привести к уменьшению потерь времени на обучение, а значит и к уменьшению потерь энергии за весь период работы.

Если дополнить такую обучающую систему свойством автоматического отключения от сети через $n+2$ тактов после последнего интервала взаимодействия, то расход энергии этой системой и вовсе станет минимальным.

Очевидно, что отслеживание управляющей системой инварианта взаимодействия (интервала взаимодействия) между машиной и человеком будет не чем иным, как синхронизацией такого взаимодействия. Откуда следует, что выдвинутый нами тезис [10] о том, что синхронизация взаимодействия человека с машиной приведет к построению системы с минимальными потерями, можно считать доказанным.

Макет АСУ обучением. Соображения, сформулированные выше, послужили основой для построения макета АСУ обучением. Он строился как макет системы, обучающей английскому языку.

Обучаемому предлагается осуществить перевод фразы, которая выводится на экран дисплея, с русского языка на английский (а не наоборот!). Перевод должен быть введен в ЭВМ посредством клавиатуры и высвечен на экране строкой ниже:

1. Это собака
2. This is a dog
- 3.

У обучаемого есть возможность обратиться к справочной информации по грамматике и к словарю по «Help».

Если обучаемый не в состоянии осуществить перевод (по незнанию либо по недостатку времени), то после достаточно большой паузы на третью строку экрана выводится верный перевод:

1. Это кошка
2. This is a
3. This is a cat

После паузы, достаточной для того, чтобы обучаемый ознакомился с переводом и запросил необходимые справки, система выводит вторую русскую фразу для перевода и т. д., пока не будет просмотрен весь учебный программный блок. Таким

образом, осуществляется передача информации и сравнение ее с эталонной, контрольной.

При этом обучаемый может опережать систему и ограничивать интервалы взаимодействия посредством нажатия клавиши ВК. Система запоминает величину интервала взаимодействия и автоматически осуществляет ее коррекцию по следующему простому алгоритму:

если обучаемый не уложился в предложенный интервал времени, то интервал увеличивается на некоторое значение Δt ;

если обучаемый уложился в предложенный интервал, но совершил хотя бы одну ошибку, то интервал увеличивается на некоторое значение Δt ;

если обучаемый уложился в предложенный интервал, т. е. успел нажать клавишу ВК, и не совершил ни одной ошибки, то зафиксированный нажатием ВК интервал запоминается системой и предлагается при следующем предъявлении.

Посредством такого простого алгоритма коррекции величины интервала взаимодействия, а также через установку цели — потребного уровня готовности памяти — представляется возможным, осуществлять управление по критериям точности и скорости воспроизведения. Уже сейчас можно сказать, что алгоритм управления может быть усложнен, поскольку если в самом начале обучения его простота (а стало быть и грубость управления) может устраивать, то в дальнейшем, при приближении к предельной (целевой) готовности памяти, он должен оценивать все более и более тонкие реакции обучаемого и проводить соответственно более тонкое управление.

После просмотра всего программного блока система осуществляет подачу тех же записей повторно, но уже с интервалами, скорректированными по приведенному выше алгоритму, — так реализуется принцип упражняемости или повторения. В отличие от традиционных систем этот принцип реализуется организованным образом: путем установления промежуточных уровней готовности памяти, которые соответствуют безошибочному и укладываемому в определенные интервалы воспроизведению английских фраз, осуществляется селекция уже освоенных для данного уровня операций по воспроизведению и удаление освоенных фраз из процесса их повторного вывода на экран, что решает вопрос организации процедуры повторения и снимает вопрос о ее неэффективности.

Обобщение и оценка результатов исследований. Изложенное позволяет в дополне-

ние к уже имеющимся рекомендациям по проектированию обучающих систем, сформулированным теорией машинного обучения, выдвинуть новые. Они составят две группы.

Первая группа — рекомендации психолого-педагогического характера:

материал, которому надлежит обучать, должен быть составлен с учетом важнейшего критерия избирательности — критерия вариативности признаков;

материал должен подаваться таким образом, чтобы перед обучаемым была поставлена задача воспроизведения образа предмета, являющегося предметом обучения.

Вторая группа — рекомендации психофизиологического уровня, т. е. относящиеся к способу функционирования управляющей системы, к возможности учета ею психофизиологических состояний обучаемого. А именно при проектировании системы необходимо реализовать в ней возможности:

учета точности и быстроты воспроизведения обучаемым предъявленного для обучения материала, поскольку точность и быстрота воспроизведения являются индикаторами уровня готовности памяти, т. е. решающим свидетельством степени прочности усвоения;

установки цели управления как уровня психофизиологического состояния, соответствующего установленному обучающим персоналом потребного уровня готовности памяти;

возобновления операции передачи (коррекции) информации в случае выявления недостаточного уровня готовности памяти обучаемого, а при достижении потребного уровня готовности памяти по отдельной операции умения осуществлять селекцию и удаление из последовательности повторений уже освоенных операций с целью устранения явления неэффективности повторения, т. е. реализовать в системе свойства управляемой упражняемости.

Вторая группа рекомендаций достаточно полно разобрана и освещена в тексте статьи, что же касается первой группы, то она, как нам кажется, требует комментария.

Критерий вариативности признаков как важнейший показатель избирательности удивительнейшим образом вписывается в данное нами определение управления. Справедливость этого становится очевидной при рассмотрении использованного нами примера:

1. This is a dog
2. This is a cat
3. This is an animal

N. This is a man

Согласно закону управления необходимо сначала создать избыточность информации, т. е. ее разнообразие, что и выражается в вариативности приведенных фраз. Однако при всем разнообразии и вариативности приведенных фраз одно в них неизменно (является инвариантом) — конструкция This is..., которую обучаемый выделяет, редуцирует из всего разнообразия, избыточности информации. Так в психолого-педагогической организации материала осуществляется наш фундаментальный закон управления обучением — управление передачей знаний на психолого-педагогическом уровне.

Вторая же рекомендация проистекает из требования теорией неперменного воспроизведения материала предмета обучения, предназначенного для усвоения. И если при обучении языку побуквенное воспроизведение фразы представляется обязательным, что особенно принципиально для обучения английскому языку, где звучание слова и его написание сильно разнятся, то как быть в случае «вопросно-ответных» обучающихся систем?

В этом случае наши рекомендации таковы: ответ на вопрос, заданный машинной, можно выбрать пословно из выводимого в нижней части экрана набора слов, содержащего слова для требуемого ответа. Этот выбор можно осуществить маркированием необходимых слов, после чего система сама автоматически осуществит сбор слов в необходимое предложение и выведет его на экран.

Заключение. Итак, проведенное исследование доказало выдвинутый нами тезис [10] о том, что синхронизация взаимодействия человека с машиной в человеко-машинных системах приводит к построению систем с минимальными потерями. Очевидно, что минимальность потерь надо понимать в феноменологическом смысле. Привлечение новых, все более утонченных приемов управления, действующих обратную связь, т. е. реализующих принцип воспроизведения, приведут в конечном итоге к общему уменьшению издержек, однако последние будут ограничены снизу определенным уровнем психолого-педагогического совершенства программ обучения, т. е. кода, понимаемого как упорядоченная пространственная совокупность и временная последовательность воздействий.

Совершенство же этого кода определяется искусством составлять программы обучения таким образом, чтобы материал этих обучающих программ был бы

подвергнут классификации настолько, что на регуляцию образа этого материала затрачивались бы минимально возможные время и энергия, пока не решены проблемы составления индивидуальных программ самими машинами — машинами, наделенными интеллектом. Однако в любом случае это можно будет сделать лишь с использованием синхронизации взаимодействия человека с машиной как принципа построения человеко-машинного интерфейса.

Таким образом, введение упорядоченности во времени посредством синхронизации «вносит жизнь» в систему. Система становится организацией или организмом. Это следует из утверждения Н. Н. Моисеева о том, что под организмом, следуя терминологии теории управления, следует понимать любую систему, которая не только имеет собственные цели, но и обладает определенными возможностями ее достижения [11].

Предложенная система обучения, в отличие от системы Г. Паска — системы пассивной адаптации, обладает как целью, так и возможностями достижения этой цели, являясь, таким образом, системой с активным управлением, т. е. организацией или организмом.

Очевидно, что проделанная нами работа открывает перспективы для построения совершенно нового класса автоматизированных систем — систем с минимальными потерями (информации, времени, энергии).

Из теории неравновесных систем хорошо известно, что потери имеют своим эквивалентом энтропию. Также хорошо известно, что с усложнением организации живых существ в процессе филогенеза, названного Н. А. Бернштейном обрастанием, т. е. усложнением центральной нерв-

ной системы путем наслаения слой за слоем, живые существа становились все устойчивее и устойчивее, т. е. все менее и менее энтропийными. Таким образом, полученные результаты косвенным образом указывают на то, что продолжение исследований по теме «Синхронизированный интерфейс человек-машина» открывает перспективы для построения *суперустойчивых систем* — систем, где компьютер воистину является продолжением человеческого интеллекта, значение чего трудно переоценить.

Литература

1. Большая советская энциклопедия. М.: Сов. энциклопедия, 1974. Т. 18.
2. Волков А. М., Михадзе Ю. В., Солнцева Г. Н. Деятельность, структура и регуляция. М.: Изд-во МГУ, 1987.
3. Лернер А. Я. Начала кибернетики. М., 1967.
4. Талызина Н. Ф. Управление процессом усвоения знаний, М.: Изд-во МГУ, 1984.
5. Смолянинов В. В. От инвариантов геометрии к инвариантам управления // Интеллектуальные процессы и их моделирование. М.: Наука, 1987.
6. Гаазе-Рапопорт М. Г., Поспелов Д. А. От амебы до робота: модели поведения. М.: Наука, 1987.
7. Зинченко В. П. Теоретические проблемы психологии восприятия и задачи генетического исследования // Хрестоматия по психологии. М.: Просвещение, 1987.
8. Ляудис В. Я. Память в процессе развития. М.: Изд-во МГУ, 1976.
9. Автоматизация производства и промышленная электроника. М.: Сов. энциклопедия, 1963. Т. 2.
10. Варданян И. А. О необходимости синхронизации в АСУ / Деп. ВНИИНТИ 29.09.88, № 7219-B88.
11. Моисеев Н. Н. Алгоритмы развития. М.: Наука, 1987.

ИВЦ

Ассоциация пользователей компьютеров «Apple»

Предлагаем большой выбор переводов статей из зарубежной периодики с программами игр, редакторов, утилит для ПЭВМ, совместимых с «Apple».

Полная поддержка и сопровождение пользователей ПЭВМ «Apple II», «Агат», «Правец».

Каталог вы можете получить бесплатно, написав по адресу: 127057, Москва, а/я 506.

Не забудьте вложить конверт с обратным адресом.

А. ГЕЙН, А. СЕНОКОСОВ

Программно-методический комплекс для классов с углубленным изучением ОИВТ

Не будем повторять те общие методологические и методические установки, которые характерны для свердловских учебников информатики ([8], [9] и др.). Рассмотрим более подробно только один подучебник, посвященный основам алгоритмизации, а для других ограничимся аннотацией. (Далее он будет называться «учебник».)

В рассматриваемом подучебнике мотивация к изучению алгоритмов создается апелляцией к опыту учащихся и ролевой игрой ([9] с. 29). Существенно больше, чем в обычном учебнике, дается акцент на понятие исполнителя алгоритмов. Практически ни один алгоритм не рассматривается без связи с тем исполнителем, для которого он составлен.

В учебниках информатики «водораздел» между алгоритмом и программой либо отсутствует вовсе ([14] с. 102; [4] с. 25), либо сводится к проблемам синтаксической правильности (действия в алгоритмах можем записывать в произвольной форме, а машинные команды — в строгом синтаксисе [3]). На самом деле — и это вторая отличительная черта учебника — при программной реализации алгоритмов мы сталкиваемся как минимум еще с двумя проблемами: ограниченностью разрядной сетки и ограниченностью объема памяти.

Есть и третья проблема, которую Д. Кнут называет проблемой «разумной конечности» алгоритма ([10], с. 32): нам важна не только принципиальная возможность получения результата за конечное время, но и возможность получе-

ния результата за разумное время. Разумеется, об этих проблемах говорится вскользь и в существующих учебниках. Это правильно: обычному пользователю готовых программных средств нет нужды вникать в тонкости таких проблем. Другое дело — подготовка профессионалов. Эти проблемы выделены в отдельную сквозную линию, которая начинается с формулировки указанных проблем в терминах «исполнитель алгоритмов». Можно сказать, что каждая из таких проблем — результат собственной интерпретации исполнителем допустимого действия. Иными словами, разные исполнители могут иметь одинаковые допустимые действия, но выполнять их каждый по-своему (скажем, на разных языках программирования один и тот же алгоритм выполняется по-разному; отсюда, в частности, и проблема выбора языка, обеспечивающего более эффективное решение задачи).

Особенно наглядно это можно показать, если в качестве исполнителя выступает человек: действительно, трудно найти двух людей, которые одно и то же действие выполняли бы одинаково.

Вот как на начальном этапе объясняется различие между понятиями «программа» и «алгоритм».

«Прочитайте внимательно две программы:

<i>How to make good tea:</i>	<i>Как приготовить хороший чай:</i>
Bring fresh water to the boil	Вскипятите свежую воду.
Warm tea-pot by rinsing out with hot water.	Ополосните заварочный чайник крутым кипятком.
Put one teaspoonful of tea per cup into the tea-	Положите чай в заварочный чайник из рас-

pot and pour immediately the boiling water into the tea.
Stir the tea after 3—5 minutes.
Add sugar to taste.

чета одну чайную ложку на чашку и сразу же залейте кипятком.
Через 3—5 мин размешайте.
Добавьте сахар по вкусу.

Левая программа, очевидно, написана для Джонни-из-за-Океана. Правая — для его русского друга Вани. Ясно, что каждый из них будет выполнять одни и те же действия, и поэтому, наблюдая за их работой, мы не сможем отличить одного от другого. Ведь фактически они выполняют один и тот же набор действий. Только записаны эти наборы для каждого на своем языке.

Более того, можно заснять на видеокассету все манипуляции Джонни или Вани, а затем показать любому жителю Земли, и ему будет ясно, как заваривать чай.

Поэтому когда говорят о самой последовательности действий для достижения какой-либо цели, то используют термин не ПРОГРАММА, а АЛГОРИТМ. Разумеется, не исполнитель должен уметь производить все действия, встречающиеся в алгоритме. Про такие действия говорят, что они ДОПУСТИМЫ для данного исполнителя. Чтобы алгоритм стал программой для конкретного исполнителя, нужно все действия, входящие в алгоритм, описать командами из его Системы команд.

Итак, алгоритм — это последовательность действий, допустимых для исполнителя, а программа — запись алгоритма на языке этого исполнителя. Это весьма тонкое различие, однако для поиска ошибок в программе даже есть специальный термин — ОТЛАДКА. Поиск ошибок в алгоритме никак по-особенному не называется».

Здесь фактически введен термин «алгоритм» и намечено первое различие между алгоритмом и программой. Методическая тонкость, пожалуй, лишь в том, что термин «программа» появился раньше термина «алгоритм». Это вызвано, как уже объяснялось тем, что в нашем учебнике исполнитель выходит на первый план, а всякий исполнитель исполняет, строго говоря, не алгоритм, а программу.

Рассмотрим еще один фрагмент объяснительного текста учебника.

«После напряженных раздумий настала пора пить чай, тот, что приготовили Джонни и Ваня. К нашему величайшему изумлению он оказался разным: от Ванино чая язык прилипает к губам, а чай Джонни почти несладкий. Почему же один и тот же алгоритм в разном исполнении привел к столь различным результатам?

Очевидно, все дело в последнем действии: добавление сахара по вкусу. Это, конечно, допустимое действие для каждого исполнителя. Просто у каждого свой вкус. А нам урок: один и тот же алгоритм, претворяемый в жизнь различными исполнителями, может давать различные результаты».

Этот шуточно оформленный пример — только начало линии, связанной с проблемами программной реализации алгоритмов. Она закрепляется и углубляется решением задач. Вот два примера.

Задача 1.

а) Петя и Коля в роли исполнителей выполняются с помощью микрокалькулятора следующий алгоритм.

Сложить 83,2438 и 57,6847

Полученный результат умножить на 10

У Пети на табло калькулятора помещается 6 цифр, а у Коли — 8. Будут ли у них одинаковые результаты исполнения этого алгоритма?

б) Приведите другие примеры алгоритмов, результаты исполнения которых зависят от их исполнителей.

Задача 2. Однажды к Пете и Коле пришел их приятель шестиклассник Саша и сказал, что им задали трудную задачу: «Для двух чисел 999/1000 и 1000/1001 установить, равны ли эти числа, и если равны, то их надо перемножить, а если нет, то из большего вычесть меньшее». Чтобы решить эту задачу, Петя и Коля составили такой алгоритм.

Разделить 999 на 1000 и результат обозначить буквой b

Разделить 1000 на 1001 и результат обозначить буквой c

Если $(b=c)$, то {умножить b на c }

Если $(b>c)$, то {из b вычесть c }

Если $(b<c)$, то {из c вычесть b }

а) Проверьте, правильно ли составлен алгоритм?

б) Напомним (см. задачу 1), что Петя и Коля в роли исполнителя выполняют алгоритмы с помощью микрокалькулятора, при этом у Пети на табло калькулятора помещается 6 цифр, а у Коли — 8. Будут ли у них одинаковыми результаты исполнения этого алгоритма?

в) Устроит ли Сашу хотя бы один из полученных результатов? Попытайтесь в уме решить Сашину задачу.

Третий отличительный аспект изучения алгоритмизации в нашем учебнике состоит в рассмотрении вопросов эквивалентности алгоритмов. Только введя понятие эквивалентных алгоритмов, мы можем достаточно квалифицированно обсуждать вопросы их эффективности. Сюда же надо отнести изучение неко-

торых методов анализа алгоритмов, в частности простейшие методы доказательства правильности и конечности.

Главным инструментом, позволяющим школьникам освоить алгоритмизацию, выступает реализованный на ЭВМ исполнитель Паркетчик. По внешнему виду он напоминает исполнителя Робот из [6], однако снабжен целым рядом возможностей, позволяющих усилить его методический потенциал.

Коротко рассмотрим содержание остальных подучебников.

Один из них посвящен введению понятия информации и ее измерению. Фактически это методологическая основа всего курса. Поэтому именно здесь рассматриваются проблемы организации информационных потоков и роль информации в жизни общества.

Подучебник «Принципы работы вычислительной техники» знакомит в первую очередь с программистскими аспектами архитектуры фон Неймана. Даются самые необходимые сведения об элементной базе ЭВМ, позволяющие хотя бы вчерне понять, как работает компьютер, построенный на этих принципах.

Главной частью учебника является знакомство с виртуальной ЭВМ «Малютка». Несмотря на свой учебный характер, она имеет набор команд и оперативную память (ОП), вполне достаточные для решения довольно-таки сложных задач. Таких, например, как игра «Ним».

«Малютка» — одноадресная машина, и для работы с ней необходимо хорошо знать 16-ричную и двоичную системы счисления, так как существует возможность в любой момент отобразить на экране содержимое ОП и изменить его. Более того, само программирование представляет из себя не что иное, как непосредственное заполнение оперативной памяти информацией. Кроме того, постоянно видны все регистры машины, благодаря чему становится наглядным даже сам процесс вычислений.

Учебная виртуальная ЭВМ является мощным мотивационным инструментом при изучении двоичной и 16-ричной арифметик, неймановской архитектуры, системы машинных команд, которая подобрана так, что представляет собой

подмножество основных команд подавляющего большинства современных «больших» процессоров.

Ассемблер виртуальной ЭВМ позволяет понять, каким образом довольно-таки простая с точки зрения логики аппаратная часть усложняется системами программирования или системами решения пользовательских задач. Наглядно демонстрируется единство программно-аппаратного комплекса, причем информация выступает зримой величиной, усложняющей организацию системы.

Опыт работы с подучебником показывает, что после его изучения становятся вполне доступными фирменные руководства по архитектуре современных процессоров и ассемблерам.

Следующий подучебник рассматривает основные типы алгоритмов с переменными. Он содержит материал, относящийся к разделам 4.1 и 4.3 содержания. Отметим, что мы не считаем необходимым строгое математическое обоснование вычислительных методов (скажем, метода Монте-Карло), а полагаем достаточным иметь убедительную иллюстрацию в сочетании с хорошими вычислительными примерами.

Технология разработки программного обеспечения — краеугольный камень в подготовке профессиональных программистов. Пренебрежение этим аспектом в совокупности со значительным машинным временем, доступным учащимся, приводит к массовому производству хакеров, совершенно не способных работать по заказу, тем более в программистском коллективе. Компьютер в руках таких подростков зачастую становится лишь средством самовыражения, а отнюдь не средством производства обогащенной информации.

Стержнем технологии разработки программного обеспечения является идеология нисходящего проектирования программного продукта. Это еще одна сквозная линия курса. Однако визуализация процесса нисходящего проектирования чрезвычайно сложна. Она требует высокопрофессионального подхода.

Именно поэтому такая работа при непосредственной поддержке академика Н. Н. Красовского ведется в Институте математики и механики Ураль-

ского отделения Академии наук СССР, стоящем у самых истоков компьютеризации школ Свердловской области.

Созданная научным коллективом под руководством В. В. Прохорова система «ПИ» («Пифагор») позволяет прямо на экране компьютера провести разработку программного комплекса методом нисходящего проектирования.

Мало того, что «ПИ» формирует образ мышления будущего профессионального программиста в единственно верном направлении. Эта система позволяет вообще абстрагироваться от конкретных процедурно-ориентированных языков: окончательный уровень разработки программного проекта с помощью конвертора может быть переведен в программу на таких языках, как Си или Паскаль.

24 Это отнюдь не означает, что в дальнейшем учащемуся не понадобится изучение процедурно-ориентированных языков. Но переход к работе на конкретных диалектах любых таких языков на любых машинах осуществляется в считанные дни.

Поэтому ясна и основная направленность подучебника, ориентированного на разработку программного обеспечения с помощью системы «ПИ».

Осталось заметить, что отсутствие IBM-совместимых машин отнюдь не является причиной отказа от изучения подучебника. Просто всё, что делается на экране дисплея в системе «ПИ», профессиональный программист обязан делать на листочке бумаги (если он хочет создать качественный программный продукт). В конце концов люди писали книги и без текстового редактора, и без персонального компьютера с принтером.

Разработка пользовательского интерфейса выделена в отдельный подучебник потому, что это проблема во многом психологическая. Профессиональный программист в своей работе столкнется с людьми самых разных профессий. Тем не менее он должен уметь написать работоспособный и удобный в использовании программный продукт для любого заказчика. Как известно, прикладная программа — это формализация знаний специалиста в конкретной области деятельности. Таким образом,

профессиональный программист должен ухитриться «пропустить через себя» знания конкретного специалиста, заказавшего ему программу. Не будет большим преувеличением сказать, что программист должен стать специалистом, пусть и ограниченным, в области деятельности, для которой пишется программа. На это ему отводятся в лучшем случае несколько недель. Совершенно очевидно, что успешное решение такой задачи требует умения работать с заказчиком, чему в основном и посвящен подучебник. В качестве практики учащиеся совместно с каким-либо учителем пишут обучающую программу по конкретному предмету. Этим оправдывается включение в учебник большого раздела, посвященного педагогическим программным средствам.

Особую роль играет подучебник, посвященный компьютерному моделированию. Вводя учащегося в основы постановок задач и построения моделей, он начинает еще одну сквозную линию учебника.

После всего вышесказанного понятно, что программно-методический комплекс для классов с углубленным изучением информатики и основ вычислительной техники — чрезвычайно сложный наукоемкий проект, требующий для своего полного завершения нескольких лет.

Тем не менее, как уже говорилось, любой из подучебников или программных компонентов может с большим успехом использоваться сам по себе, вне связи с комплексом.

К сегодняшнему дню полностью реализованы все компоненты комплекса (кроме книги для учителя), обеспечивающие преподавание информатики в первые полтора года обучения (4 ч в неделю, база — семь классов общеобразовательной школы).

Работа над комплексом продолжается!

По вопросам приобретения отдельных компонент комплекса и комплекса в целом обращаться:

620137, Свердловск, Июльская, 32,
Центр программирования «Маугли», тел.:
(8-3432) 49-60-66, (8-3432) 41-84-02,
Сенокосов Александр Иванович.

Факс: (8-3432) 57-92-56 Гейну А. Г.

Литература

1. Матюшкин-Герке А. Школьная информатика: уроки, проблемы, перспективы // ИНФО. 1991. № 1. с. 99—110.
2. Сапир М. Курс ОИВТ — каким мы его видим // ИНФО. 1990. № 5. С. 31—33.
3. Гейн А., Житомирский В., Линецкий Е., Сапир М., Шолохович В. Основы информатики и вычислительной техники. М.: Просвещение, 1991. 254 с.
4. Каймин В., Щеголев А., Ерохина Е., Федюшин Д. Основы информатики и вычислительной техники. М.: Просвещение, 1989. 272 с.
5. Кушниренко А., Лебедев Г., Сворень Р. Основы информатики и вычислительной техники. М.: Просвещение, 1990. 224 с.

6. Романов В. Информатика и прикладная математика // ИНФО. 1987. № 6. С. 35—42.
7. Гейн А. Г., Сенокосов А. И. Программа по основам информатики и вычислительной техники для специализированных классов // Компьютерные технологии в учебно-воспитательном процессе школы и вуза: Сборник. Свердловск, 1990. С. 92—107.
8. Гейн А., Житомирский В., Линецкий Е., Сапир М., Шолохович В. Программно-методический комплекс по курсу школьной информатики // ИНФО. 1988. № 3. С. 11—24.
9. Гейн А., Линецкий Е., Сапир М., Шолохович В. Информатика: модели, алгоритмы и исполнители // ИНФО. 1989. № 3. С. 25—33.
10. Кнут Д. Искусство программирования для ЭВМ. Т. 1. М.: Мир, 1976. 735 с.

П. БРУСИЛОВСКИЙ, Т. ГОРСКАЯ-БЕЛОВА, М. ЗЫРЯНОВ

Интеллектуальная компьютерная среда для обучения физической географии

Огромные возможности компьютеров, как правило, используются в обучении не полностью. Это происходит из-за того, что компьютер пытаются приспособить к существующим содержанию и формам обучения.

К сожалению, на наш взгляд, для многих предметов пока еще не ясен механизм, позволяющий эффективно включить компьютер в учебный процесс. В данной статье мы предлагаем один из возможных «компьютерных» подходов к обучению географии, основанный на методе моделирования. Для проверки этого подхода нами был построен «компьютерный» фрагмент курса физической

географии. Ход обучения для этого фрагмента поддержан интеллектуальной компьютерной средой, которая подробно описана в статье. При разработке среды авторы пытались использовать возможности современной аппаратуры (графика, цвет, звук), современный интерфейс (окна, меню, пиктограммы) и опыт построения интеллектуальных систем (экспертные системы).

Компьютерное моделирование и обучение географии

Общие идеи моделирования, как универсального подхода к изучению сложных взаимосвязей внутри предметной области, уже давно и заслуженно завоевали популярность в таких отраслях знания, как математика, физика, экономика. Советский философ В. Штофф дает такое определение модели: «Под моделью понимается такая мысленно представляемая или материально реализованная система, которая, отображая или воспроизводя объект исследования, способна замещать его так, что ее изучение дает нам новую информацию об этом объекте» [1].

Согласно В. В. Давыдову [2], с точки зрения использования моделей в обуче-



нии, важны четыре основные характеристики модели:

1. Модель — средство познания.

2. Модель — представитель оригинала, который в каком-либо отношении более удобен для изучения и может при этом перенести полученные знания на объект.

3. Модели характеризуются существенными свойствами прототипа.

4. Модели однозначно соответствуют оригиналу.

Процесс моделирования может рассматриваться как движение от сложной, многосторонней реальной действительности к ее модели, где эта действительность заведомо обеднена, но зато подчеркнуты основные связи, главные процессы, виден механизм развития. Процесс размоделирования, т. е. от модели к реальности, идет уже с новыми знаниями, которые позволяют глубже понять сущность реальности.

26 Моделирование определяется нами как основной метод приобретения знаний. Работая с моделью, учащийся поэтапно познает представленный ему объект, но постепенно рамки модели становятся ему «тесны», он понимает ограниченность предложенной модели, тогда возникает потребность в следующей модели и т. д. Двигаясь от модели к модели, дети понимают «архитектуру» науки, задача только в том, чтобы правильно представить им это движение. Таким образом, курс должен строиться как расширяющаяся система взаимодействующих моделей.

Вниманию читателей предлагается разработка моделирующей учебной среды «Остров» для VII класса (курс «Физическая география материков и океанов»), которая является практическим результатом нашего изучения проблемы моделирования на компьютере, с учетом рассмотренных выше подходов. Цель использования среды — осуществить переход от разрозненных и упрощенных понятий физической географии к компьютерному моделированию их существенных взаимосвязей в условиях единого природного комплекса.

Содержание работы в среде заключается в создании модели, иллюстрирующей структуру и основные особенности



природного комплекса на примере любого острова, произвольно выбранного в северной части Атлантического океана. Остров, как объект исследований, выбран не случайно. Он является единицей, объединяющей все компоненты природы и представляющей их взаимосвязь. Для упрощения остров рассматривается как точка на карте, а его природный комплекс определяется как единый для всей его площади.

Опираясь на методику минимизации используемых понятий для создания модели, мы пришли к выводу, что основополагающими условиями формирования природного комплекса острова можно признать следующие географические компоненты:

1. Географическое положение острова.
2. Строение дна океана в районе острова.
3. Тип земной коры.
4. Происхождение острова.
5. Тектоническое строение острова.
6. Рельеф острова.
7. Климат.
8. Почвенно-растительный покров.
9. Животный мир.

Каждый компонент может иметь несколько возможных значений. Естественно, что значения отдельных компонентов тесно связаны между собой. Согласно сценарию работы в среде, ученик как бы оказывается на необитаемом острове, географическое положение которого он видит на карте, изображаемой на экране. Зная положение острова, ученик должен определить в соответствии с географическими фактами и

закономерностями значения перечисленных выше компонентов природы острова.

Структура компьютерной среды

Компьютерная среда представляет собой несколько связанных систем, реализованных на персональной ЭВМ типа IBM PC. Каждая из систем реализует некоторый сценарий и стиль обучения. В начале работы с любой из систем ученик, система или педагог (в зависимости от вида сценария) выбирает остров в северной части Атлантического океана. Далее каждая из систем дает ученику возможность определить в процессе диалога значения всех компонентов природного комплекса и проверить правильность определения компонентов.

Основные возможности системы можно продемонстрировать на примере простейшего «контрольного» сценария диалога. В начале диалога система выдает ученику «Компьютерную карту», на которой отмечено положение острова. Вместе с каждым вопросом система предлагает меню возможных значений запрашиваемого компонента, выбирая из которого ученик дает ответы.

Система: *Определите строение дна океана*

Ученик: Материковая отмель (платформа)

Система: *Определите тип земной коры*

Ученик: Океаническая

Система: *Определите происхождение острова*

Система проверяет выбранные обучаемым значения компонентов:

Тип земной коры океаническая — ошибка!

Если строение дна океана — материковая отмель (платформа), то тип земной коры — материковая.

Реакция системы на действия обучаемого не задана заранее, как в обычных системах обучения с выборочным ответом. Для любого положения острова и для любого выбранного множества значений природных компонентов система может детально проверить и прокомментировать выбор ученика. Такие возможности дают системе встроенные в нее

«знания» о закономерностях строения природных комплексов.

Основу базы знаний системы составляет набор так называемых правил, связывающих значения разных природных компонентов. Эти правила принято представлять в форме ЕСЛИ — ТО, например «ЕСЛИ строение дна океана — материковая отмель (платформа), ТО тип земной коры — материковая». Кроме того, в системе есть простейшая «машина вывода», которая умеет применять правила к имеющимся данным (выбранным значениям компонентов), выстраивать данные в логические цепочки, делая тем самым выводы из имеющихся данных. Так, например, зная, что строение дна океана в районе выбранного острова — материковая отмель (платформа) и пользуясь правилами из базы знаний...

«ЕСЛИ строение дна океана — материковая отмель (платформа), ТО тектоническое строение острова — платформа.»

«ЕСЛИ тектоническое строение острова — платформа, ТО рельеф острова — равнина.»

...машина вывода может установить, что рельеф выбранного острова — равнина.

Машина вывода может работать как в прямом, так и в обратном направлении. То есть, зная, что рельеф выранного острова — равнина, и не имея данных о его тектоническом строении, система может установить, что тектоническое строение острова — платформа.

База знаний и машина вывода образуют простейшую экспертную систему (подробнее об экспертных системах можно прочитать в [3]), которая и является сердцем среды «Остров».

Знания в среде «Остров» используются в двух основных режимах: демонстрационном и контрольном. В демонстрационном режиме система может давать примеры применения знаний для прямого и обратного вывода: при задании учеником или учителем климата острова показать, какие животные на нем могут встречаться (следствие); наоборот, если задано, что рельеф острова равнинный, сообщить, каково должно быть его происхождение (причина). В контрольном режиме система может проверять, пра-

вильно ли обучаемый определяет компоненты выбранного на карте острова. Для этого система решает ту же задачу вслед за учеником и сравнивает его ответы со своими. При такой технологии проверки система может не только оценить ответ ученика как верный или неверный, но даже проверить правильность сделанного логического вывода и даже определить, незнание каких правил вызвало наблюдаемую ошибку ученика. По сути дела проверяется знание обучаемым закономерностей строения природного комплекса острова и умение мыслить логически, применяя эти знания.

Несмотря на то что правила хранятся в базе знаний во «внутреннем» формате, удобном для обработки, система может генерировать выводы, сообщения об ошибках и пояснения на естественном языке (ЕЯ). Так, если ученик верно определил вулканическое происхождение острова и неверно — материковый тип земной коры, система сообщит буквально следующее: «Рельеф острова низкие горы — ошибка! Если тектоническое строение острова — платформа, то рельеф острова — равнина». Это сообщение (как и все ЕЯ-сообщения системы) может быть сгенерировано на любом из нескольких ЕЯ (русском, английском и т. д.), поддерживаемых системой. Для того чтобы настроить среду для работы с новым языком, достаточно сформировать (переписать) текстовые файлы словаря и сообщений, после чего программа будет «разговаривать» с учеником на его языке. Текстовый формат файла сообщений позволяет также подыскать более подходящие фразы, заменив ими неудачные; такая возможность, вероятно, будет полезна учителям.

Интерфейс с учеником в среде «Остров»

Интерфейс системы использует современные черты человеко-машинных интерфейсов: пиктограммы, меню, графику, цвет, звук, многооконный режим. Выбор географического положения острова может проводиться с помощью экранной компьютерной карты, на которой изображена северная часть Атлантического океана. Текущее положение отмечено на карте графическим курсором. Пере-

двигая курсор по карте с помощью клавиш со стрелками, можно выбрать любое желаемое положение острова.

Для выбора значений компонентов острова используются пиктографические и текстовые меню. Каждому возможному значению основных компонентов природного комплекса сопоставлена принятая для него в географической науке условная картинка — пиктограмма. При выборе учеником значения конкретного компонента ему выдается набор пиктограмм, представляющих все возможные значения этого компонента. По этим пиктограммам можно перемещаться с использованием клавиш со стрелками, как по обычному меню. В демонстрационном режиме картинки сопровождаются соответствующими подписями.

С использованием пиктограмм строится также так называемый экран состояния, который доступен ученику в любой момент работы в системе. На экране состояния отображается уменьшенная карта с отмеченным положением острова и выводится по одной пиктограмме для каждого компонента. Для тех компонентов, значение которых уже выбрано учеником к этому моменту, выводится пиктограмма этого значения, для остальных выводится знак вопроса. Экран состояния дает ученику наглядную информацию о состоянии работы. Кроме того, он используется для наглядной демонстрации ошибок ученика после проверки: пиктограммы неверно выбранных значений просто перечеркиваются крест-накрест.

Учебные сценарии

Возможности среды позволяют реализовывать для разных целей разные учебные сценарии работы в виде отдельных систем. К настоящему времени реализовано два сценария работы: «исследовательский» и «контрольный». В «исследовательском» сценарии работой в системе управляет ученик. Задав самостоятельно по карте расположение острова, он может далее фиксировать в любом порядке значения компонентов, запрашивать в любой момент состояние своего выбора и проверку его правильности, изменять свой выбор и снова запрашивать проверку. Исследовательский сценарий дает

ученику среду для самостоятельной работы и обучения в режиме экспериментирования «смотри и пробуй», который становится все более популярным в учебных системах. Этот сценарий может также применяться учителем с целью демонстрации в ходе объяснения или повторения материала.

Контрольный сценарий предназначен для контроля и оценки знаний. Здесь работой ученика управляет система. В строгом порядке она просит обучаемого задать на карте положение острова и последовательно выбрать из меню значения всех его компонентов. В заключение система проверяет сделанный выбор, демонстрирует ошибки на экране состояния, генерирует сообщения об ошибках и пояснения на ЕЯ и выставляет оценку. От простейшего контрольного сценария, пример которого приведен выше, реализованный сценарий отличается несколькими особенностями. Во-первых, введен дополнительный контроль знания пиктограмм. Пиктографическое меню не содержит подписей, и после выбора картинки обучаемый сам должен подобрать к ней подпись из меню. Во-вторых, для придания работе игрового характера на экране ведется подсчет очков, начисляемых за правильный ответ и за правильный выбор картинки. В-третьих, обучаемый может использовать трехуровневую помощь-подсказку. Сценарий помощи задается в отдельном файле, что позволяет учителю, использующему систему, гибко «программировать» эту помощь. Конечно, при использовании подсказкой количество очков за правильный ответ снижается. Как подсказка, так и экран состояния могут быть вызваны в любой момент работы.

Предоставление помощи

Для более гибкого задания консультационной помощи создана многоуровневая подсистема и разработан язык описания сценариев консультаций (ЯОСК). Подсистема работает в режиме интерпретации ЯОСК.

Сценарии консультации представляют собой набор кадров сценария. Каждый кадр сценария соответствует некоторому значению одного из компонентов природного комплекса острова и задает структуру подачи консультационного ма-

териала при определении компонентов острова. Количество уровней консультации и их содержание полностью определяются содержанием кадров. Особенности подсистемы помощи в том, что при подсчете набранных баллов оцениваются только первые три уровня.

Подсистема позволяет выводить на экран тексты и картинки, устанавливать размеры окна вывода, выбирать нужный цвет букв. Возможно также синтезировать на естественном языке объяснение подсистемы логического вывода, осуществлять выбор действий в зависимости от некоторого условия, предложить обучаемому самому сделать выбор, какую информацию он еще хочет получить. При необходимости можно начать выполнение другого кадра или закончить подачу материала, не дожидаясь конца работы текущего.

Перечисленные возможности ЯОСК, как надеются авторы, будут полезны учителям и методистам и позволят настроить систему на конкретный класс, на конкретных учеников.

История создания и перспективы развития

● Первый «исследовательский» сценарий для среды «Остров» был закончен и опробован в рамках Международной компьютерной школы 1989 г. в г. Дубне [4]. В коллектив разработчиков кроме авторов, входили психолог В. Хозиев (МГУ), а также словацкие, немецкие, болгарские и советские школьники. В набор поддерживаемых системой ЕЯ входили языки всех перечисленных стран и английский язык. «Контрольная» версия системы завершена в 1990 г. на факультете ВМК МГУ М. И. Зыряновым и практически используется на уроках географии в московской школе № 243. Желаящие получить (безвозмездно) эту версию для использования ее в учебном процессе могут обращаться к авторам по телефону 281-24-93 (Татьяна Борисовна Горская-Белова).

Среди ближайших планов по развитию системы «вширь» — создание «игрового сценария» с подсчетом очков и таблицей рекордов. Здесь обучаемый за игру должен будет «решить» три слу-

чайно выбранных на карте острова.

Для развития системы «вглубь» ее предполагается дополнить фактографической информацией. Кванты информации будут связаны с конкретными значениями компонентов природного комплекса и с отдельными правилами. Планируется также дополнить систему моделью знаний ученика. В этой модели индивидуально для каждого ученика будет накапливаться информация о знании/незнании им отдельных правил и пиктограмм. Модель сможет использо-

ваться самой системой для адаптации своей работы к конкретному ученику, а также преподавателем для понимания картины знаний конкретного ученика.

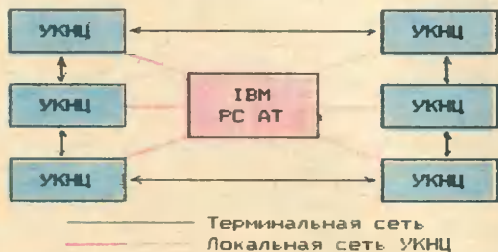
Литература

1. Штофф В. Моделирование и философия. М.: Наука, 1966.
2. Давыдов В. В. Проблемы развивающего обучения. М.: Педагогика, 1986.
3. Попов Э. В. Экспертные системы. М.: Наука, 1987.
4. Международная компьютерная школа: проблемы и перспективы // ЭВМ в образовании. Педагогические программные средства. Рига: ЛатГУ, 1989. С. 5—28.

НОВЫЕ ПРОЕКТЫ

Многопользовательский персональный компьютер

В результате разработки фирмы «Технология» стало возможным использование ПК «Электроника МС 0511» в качестве терминала многопользовательской системы на основе персонального компьютера IBM PC/AT. Система представляет локальную сеть звездообразной структуры, в центре которой — PC AT/286 (AT/386), а в лучах «звезды» — до 16 терминалов (рис. 1). В результате пользователи терминалов УКНЦ (ученики) получают доступ к ресурсам и возможностям профессиональной ПЭВМ семейства IBM PC. Фактически на центральном ПК этой локальной сети работает несколько виртуальных процессоров, поддерживающих среду MS DOS, для которых ввод/вывод переориентирован на терминалы УКНЦ.



Внешним эффектом такого решения становится почти «бесплатный» доступ пользователей УКНЦ к широкому кругу программного обеспечения, работающего под MS DOS на IBM-совместимых ПК в алфа-

витно-цифровом режиме. Важно отметить, что при разработке прикладного программного обеспечения для данной системы можно использовать мощное инструментальное программное обеспечение IBM PC/AT (отладчики и компиляторы, включая объекто-ориентированные языки программирования и разнообразные пакеты программ) и последние достижения (языки 4-го поколения и т. д.). Прямым следствием этого является значительное сокращение времени создания прикладного программного обеспечения. Например, разработка программы редактирования текстов (типа «Лексикон») будет занимать считанные дни. При этом сохраняется возможность работы стандартной локальной сети кольцевого типа КВУ «Электроника МС 0202» и программного обеспечения, разработанного для этого режима работы УКНЦ.

Более того, центральная машина в терминальной сети ЭВМ программно совместима с IBM PS/2, используемой в проекте «Пилотные школы». Это означает, что программное обеспечение, разработанное для «Пилотных школ», может быть использовано на центральной ЭВМ без каких-либо ограничений, но в этом случае терминальная сеть не используется. Технически принципиально возможно адаптировать ПО «Пилотных школ» и под терминальную сеть, но это потребует дополнительных усилий, возможно сравнимых по трудозатратам с созданием

нового программного обеспечения. Для подключения существующих классов УКНЦ к IBM PC/AT не требуется внесения конструктивных технических изменений в ПК «Электроника МС 0511» (обмен с центральной ПЭВМ PC/AT происходит через интерфейс С-2 УКНЦ). Пользователь в момент начала работы с терминалом может выбрать режим работы: в локальной сети КВУ УКНЦ либо в терминальной сети PC/AT (естественно, в любой момент можно изменить режим работы, произведя перезагрузку терминала). Таким образом, в одном классе УКНЦ возможна одновременная работа учащихся как в стандартной локальной сети КВУ, так и в режиме терминала PC/AT. Техническое решение — программно-аппаратный комплекс, включающий оригинальную плату, устанавливаемую в стандартный разъем расширения PC/AT, к внешним выводам которой может быть подключено до 8 терминальных устройств (в PC можно установить до двух таких плат), и программы, обеспечивающие межмашинный обмен и управляющие виртуальными процессами на центральной ПЭВМ. Реализованы два варианта предложенной терминальной сети, принципиально различающиеся по функциональным возможностям.

Первый основан на ПЭВМ класса IBM PC/AT/286; при этом все терминалы PC/AT работают с одной программой, выполняемой на центральной ПЭВМ. Например, при работе с программой Norton Commander, или любой другой, выполняющейся в среде MS DOS, на каждый терминал транслируется одно и то же изображение экрана PC/AT и ввод с клавиатуры каждого терминала интерпретируется как аналогичное воздействие на клавиатуру центральной ПЭВМ. С другой стороны, если на PC/AT выполняется программа, специально разработанная для такой терминальной сети, то с разными терминалами центральной ПЭВМ будут работать различные подпрограммы этой программы, создавая эффект использования многозадачной системы. Необходимо отметить, что возможности первого варианта реализации терминальной сети ограничены, с одной стороны, однозадачным режимом работы MS DOS на PC/AT/286 и, с другой стороны, сложностью разработки специального программного обеспечения для поддержания мультизадачности на микропроцессоре Intel 80286.

Важно отметить, что фирма «Технология» разработала вариант мультизадачного монитора, открытого к расширению, и готова предоставить соответствующую документацию для разработчиков прикладного программного обеспечения. Концептуально монитор построен как расширение BIOS и DOS,

соблюдая все соглашения DOS. Используя монитор, разработчик может сосредоточить свои усилия на прикладных вопросах, содержательной обработке информации, не вникая в детали организации мультизадачности. В этом случае значительно снижаются требования к квалификации прикладного программиста и упрощается процесс создания программного продукта.

Второй вариант, базирующийся на ПЭВМ класса IBM PC/AT/386, существенно использует аппаратные особенности микропроцессора этого ПК для реализации многопользовательского мультизадачного режима работы в терминальной сети. При этом каждому пользователю может быть предоставлена возможность работы со своим программным обеспечением в среде MS DOS. Например, первый пользователь работает с Norton Commander, второй — с SuperCalc, третий — с СУБД dBASE IV и т. д. Количество одновременно выполняющихся задач ограничено только объемом оперативной памяти PC AT/386. Для одновременной независимой работы 12 пользователей терминальной сети необходимо не менее 4 Мбайт ОП. В зависимости от учебного курса и организации работы пользователей возможно сокращение объема оперативной памяти (но не менее 2 Мбайт).

В качестве базовой многозадачной операционной системы используется Desq View/386, адаптированная к работе в предлагаемой терминальной сети. Использование другой многозадачной многопользовательской операционной системы PC MOS/386 предоставляет пользователям терминальной сети дополнительные услуги по разграничению доступа к ресурсам IBM PC. Выбор многозадачной операционной системы определяется областью приложения терминальной сети. Заметим, что в этом случае активно используется программное обеспечение, разработанное зарубежными фирмами; создание же отечественных лицензионно чистых аналогов потребует значительных дополнительных усилий.

Необходимо отметить, что при такой реализации терминальной сети может использоваться прикладное программное обеспечение, разработанное для варианта, основанного на PC AT/286.

Достоинства и недостатки предлагаемых вариантов реализации терминальной сети. Основными преимуществами такой терминальной сети являются значительное (в 3—5 раз) удешевление рабочего места пользователя ПЭВМ семейства IBM PC и, соответственно, доступ к существующему программному обеспечению для IBM-совместимых компьютеров. Однако такой эффект достигается за счет увеличения времени реакции

программного обеспечения в диалоге с пользователем терминала из-за относительно невысокой скорости обмена (9600 бод) через стандартный интерфейс С-2 «Электроники МС 0511». Кроме того, накладываются ограничения на использование в терминальной сети программного обеспечения, написанного с нарушениями соглашений базовой системы ввода/вывода (BIOS) операционной системы MS DOS в зависимости от характера нарушений вызова функций BIOS. Впрочем, последствия некоторых нарушений можно устранить. Например, для адаптации программного обеспечения, использующего прямой доступ к видеопамяти, применяются специально разработанные в фирме «Технология» программные средства. Одной из трудностей, возникающих при адаптации программного обеспечения, является преодоление конструктивных различий в клавиатурах УКНЦ и IBM PC, решаемой переобозначением клавиш.

32

Дополнительной проработки также требует использование графического интерфейса в прикладных программах, связанное с объемом передаваемой в терминал видеоинформации. Результатом этого является значительное (в 4 раза) увеличение времени реакции в диалоге. Однако применение соответствующей методики обучения позволит в известной мере снизить влияние указанных недостатков. Аппаратные и программные средства разработанной терминальной сети открыты для развития и повышения эксплуатационных характеристик. Например, возможна замена терминалов УКНЦ на другие ПК, поддерживающие асинхронный последовательный ввод/вывод по стандарту RS232 (С-2). В частности, таким терминалом может стать ПЭВМ семейства IBM PC. В этом случае возможно использование терминальной ПЭВМ как терминала центральной ЭВМ, так и как независимой ПЭВМ, объединенной с центральной ЭВМ в локальную сеть. При этом не требуется дополнительное оборудование, так как сетевое взаимодействие организуется через стандартный интерфейс. В дальнейшем локальная сеть может быть интегрирована в региональные сети и соответственно модифицировано программное обеспечение. Также центральная ПЭВМ может быть заменена на более совершенные модели. Один из возможных вариантов — замена только материнской платы IBM PC AT/286 на AT/386.

За счет применения других алгоритмов трансляции видеоинформации в терминальные устройства может быть существенно сокращено время реакции в диалоге. Используемые терминалы могут быть оснащены координатными манипуляторами типа «мышь»

(фирма «Технология» предлагает такое расширение для УКНЦ). Но магистральный путь развития предлагаемой терминальной сети состоит в разработке специализированного программного обеспечения. С учетом архитектурных особенностей оборудования и методических приемов преподавания учебных курсов возможно создание информационно-обучающей среды, компенсирующей недостатки данной системы. Использование современных технологий и соответствующих инструментальных систем позволяет значительно сократить трудозатраты на программирование. Предоставляемая пользователям документация по программным средствам, разработанным в фирме «Технология», и широкое распространение IBM-совместимых ПЭВМ позволяют привлечь к разработке прикладных программ значительное число специалистов непосредственно в школах и т. д. Важно, что прикладное программное обеспечение для терминальной сети легко преобразуется к работе на обыкновенных IBM PC/XT/AT (IBM PS/2) на системном уровне. При этом координацию усилий и обмен информацией должны взять на себя региональные центры новых информационных технологий, возможно, на коммерческой основе. В этом случае фирма «Технология» планирует продолжать развитие системных программ и аппаратных средств, используемых в терминальной сети.

Важной чертой данного комплекса является то, что все методическое и программное обеспечение, созданное для КВУ УКНЦ (в традиционной конфигурации), может по-прежнему применяться в учебном процессе. Тем самым налицо преемственность комплекса к программам и методикам для УКНЦ, как уже созданным, так и тем, которые будут создаваться. В частности, пакеты прикладных программ, предназначенные для формирования компьютерной грамотности учащихся в процессе изучения предмета «Основы информатики и ВТ», целесообразно использовать при работе в прежней конфигурации КВУ.

Вместе с тем функциональные возможности учебных ПЭВМ УКНЦ при работе в стандартной конфигурации могут быть расширены посредством использования возможностей IBM PC. Наличие файлового сервера позволит, не прибегая к услугам не вполне надежной «стандартной» сети КВУ, осуществлять обмен информацией между компьютерами. При этом ученики, с одной стороны, работают в рамках системы соглашений, принятой для «стандартной» сети, с другой стороны, им становится доступным, в частности, использование накопителя на жестком магнитном диске. Это позволяет использо-

вать в учебном процессе содержательные и методические подходы, требующие большого объема памяти ЭВМ для хранения разработок учеников. Например, при изучении предмета «Основы информатики и ВТ» становится возможным решение крупных в объемном отношении задач путем создания пополняемых от занятия к занятию библиотек, что оказывалось затруднительным применительно к стандартной конфигурации КВУ УКНЦ. Таким образом, преемственность комплекса по отношению к «стандартной» конфигурации КВУ УКНЦ не только сохраняется, но и в рамках этой преемственности также осуществляется определенное развитие возможностей.

С другой стороны, наличие в составе комплекса компьютера IBM PC AT открывает доступ к программным средствам, разработанным для IBM-совместимой техники. Имеющиеся и разрабатываемые инструментальные программные средства позволят создавать прикладное программное обеспечение учебного назначения как для ПЭВМ УКНЦ, работающей в составе «стандартного» КВУ, так и для работы комплекса в целом. Прикладное программное обеспечение, созданное и вновь создаваемое для компьютеров стандарта PC XT/AT, может быть использовано в учебном процессе в демонстрационных целях на головной ЭВМ AT/286, входящей в состав комплекса; определенная часть программного обеспечения (а также специально адаптированное для использования в работе комплекса) пригодна для работы IBM PC в мультизадачном режиме. При этом в качестве терминалов AT используются компьютеры учеников (ПЭВМ УКНЦ).

С дидактической точки зрения представляется важным реализуемая комплексом возможность организации процесса оперативного обмена информацией между ПЭВМ учеников, координируемого IBM PC. Подобная система, поддерживающая «распределенный диалог» между учителем и учениками, по существу, является учебно-игровой средой, спектр применений которой в учебном процессе может быть достаточно широк. Это позволяет не только расширить круг тем предмета «Основы информатики и ВТ», которые могли бы изучаться с помощью вычислительной техники, но и разработать конкретные направления применения компьютерного комплекса при изучении других предметов. Проведение фронтальных проверочных работ по контролю репродуктивных знаний учеников в среде, предоставляемой данным комплексом, позволит сочетать оперативность контроля знаний с занимательной формой проведения работы. При этом по окончании проверки преподаватель получает



33

информацию о результатах контроля в удобной для него форме (например, в виде сводной ведомости). При работе в данном режиме от учителя-предметника не требуется практически никаких специальных знаний в области компьютерной техники, программирования и т. п. Подобные проверочные работы могут проводиться практически по любому традиционному или экспериментальному школьному курсу.

Помимо контроля знаний другим направлением деятельности педагога может служить формирование и активизация аудиовизуальных ассоциаций учеников, например, при освоении родного и иностранных языков. В целом же возможности, предоставляемые комплексом, позволяют ориентировать учащихся на коллективную деятельность для достижения общей цели в рамках учебно-игровой среды. Надежность и оперативность обмена данных по «сети» комплекса обеспечивают тем самым возможности, реализовать которые в рамках стандартного КВУ затруднительно.

Фирмой «Технология» планируется выпуск методических, программных и аппаратных средств, предоставляемых учителю для реализации описанных выше форм учебной деятельности.

Заслуживают внимания также возможные деловые применения комплекса в учреждениях народного образования. При функционировании комплекса допускается значительное (до 120 метров) удаление каждого терминала от головной ПЭВМ IBM PC AT.

Возможна установка терминалов «УКНЦ» (по одному) в двенадцать и более кабинетов (допускается одновременное подключение до 16 терминалов). Преподаватель или администратор, находящийся в одной из этих аудиторий, получает возможность обращения к IBM PC/AT для получения справочной информации, внесения изменений в базу данных и т. д. Деловые применения комплекса в учреждениях образования помимо непосредственного эффекта демонстрируют уча-

щимся целесообразность применения средств компьютерной техники в практической деятельности.

Таким образом, целесообразность использования комплекса как в учебном процессе, так и в административной деятельности учреждений народного образования обусловлена возможностью, наряду с применением традиционных методических подходов и программных средств, осуществлять внедрение новых образовательных технологий.

Фирма «Технология»

**предлагает учреждениям народного образования,
располагающим КВУ «Электроника МС-0202» (УКНЦ),
стать акционерами проекта
«Многотерминальный комплекс учебного назначения»
(на базе IBM PC AT)**

34

Первый этап проекта «Многотерминальный комплекс учебного назначения» реализуется на подписной (акционерной) основе. Став акционером проекта, ваша школа или учреждение получат возможность:

дооснастить КВУ УКНЦ компьютером IBM PC/AT, а также многопользовательским интерфейсом IBM PC — УКНЦ. Этот интерфейс полностью сохраняет возможность «стандартной» конфигурации КВУ УКНЦ, а значит, по-прежнему применимы все программные средства, написанные для УКНЦ. Более того, со временем IBM PC «возьмет на себя» дублирование работы не вполне надежной «основной» сети, связывающей «Электроники», и предложит пользователям УКНЦ возможность работы с накопителем на жестком магнитном диске «Винчестер». Фирма «Технология» обеспечивает сдачу комплекса «под ключ»; регулярно приобретать заинтересовавшие вас программы и программно-аппаратные средства, разрабатываемые фирмой «Технология» для «стандартной» конфигурации КВУ УКНЦ;

получать прикладные и инструментальные программные средства, предназначенные для комплекса IBM PC — УКНЦ.

Если в вашем учреждении нет КВУ УКНЦ, фирма готова поставить его и смонтировать.

Разрабатываемые фирмой подходы к организации учебной деятельности, воплощенные в предлагаемых программных средствах, позволяют:

использовать комплекс при изучении раз-

личных предметов, а также в административной деятельности учреждения образования;

перестать беспокоиться о работоспособности техники: фирма «Технология» обеспечивает обслуживание и ремонт как комплекса УКНЦ, так и комплекса в целом;

своевременно получать детально разработанные дидактические материалы для организации занятий с использованием комплекса по различным учебным предметам; повышать квалификацию сотрудников на курсах, организуемых фирмой «Технология». Вы можете заказать курсы как по тематике, связанной с использованием комплекса в преподавании конкретных учебных предметов, так и по общедидактической проблематике; получать оперативные консультации по двум «горячим линиям»:

- а) технические инновации,
- б) методические разработки;

принимать участие в конференциях и семинарах, проводимых фирмой, для определения направления дальнейших разработок.

Фирма «Технология» работает для каждого из акционеров проекта. Каждый акционер имеет право голоса.

Программа первого этапа рассчитана на 2 года.

Количество акционеров ограничено!

Если вы решили ограничиться информационным участием (стать «корреспондентом» проекта), вам достаточно внести организационный взнос.

Телефон для справок в Москве: 128-19-25. Мы ждем ваших заявок.

Предлагаем

владельцам и пользователям ПЭВМ типов «Вектор-06Ц», «Львов ПК-01», УКНЦ («Электроника МС 0511»), БК-0010.01, БК-0011, IBM XT/AT, «Электроника МС 1502», «Поиск», «Специалист», «Синклер ZX Спектрум», «Правец-8Д», ДВК-3, 4, РК-86 32К, «Микроша», «Партнер», «Апогей» «Орион», «Агат-7», «Корвет», «Атари ХЕ/ХЛ», «Сура», «Ассистент»

широкий выбор системных, прикладных, игровых учебных программ, новейшие разработки из первых рук по умеренным ценам

(средняя стоимость одной программы — 8—10 рублей!). Каталоги программ можно получить по почте или купить в кооперативе «Электрон». Для получения каталогов программ и бесплатных рекламных листов необходимо выслать в адрес кооператива «Электрон» открытку с указанием типа компьютера и своего домашнего адреса (организациям — выслать гарантийное письмо-заявку в адрес кооператива «Электрон»).

Предлагаем

учебные программы для классов УКНЦ («Электроника МС 0202»).

Закключаем

с авторами договоры на тиражирование разработанного ими программного обеспечения с выплатой процентов от реализации.

Покупаем

программы для ПЭВМ «Апогей», «Поиск», «Электроника МС 1502», «Ассистент», «Сура», «Орион»; возможен обмен программами.

Продаем

организациям по безналичному расчету ДВК всех модификаций, УКНЦ («Электроника МС 0511»), классы УКНЦ («Электроника МС 0202») с программным обеспечением.

Продаем

дополнительные контроллеры и периферийные устройства для ДВК, контроллеры КМД, КНГМД, КЖД, КГД, КСМ, КЦГД, платы микроЭВМ с ОЗУ 256К (1М), накопители на ГМД МС 5305, МС 5311, МС 5309, МС 5310, накопители «Винчестер» 5М, 10М, 20М, 40М, сетевые контроллеры ИРПС-6, цветные мониторы МС 6106.

Оснащаем

компьютеры УКНЦ («Электроника МС 0511») кассетами ППЗУ с Бейсиком, дисковыми контроллерами, КМД. Продаем стенды для диагностирования и ремонта УКНЦ.

Продаем

дополнительные платы для IBM PC/AT: платы ввода аналоговой информации (16 каналов), плата адаптера IEEE-488 (МЭК-625, КОП), плата интерфейсного адаптера PCI-1 (содержит порт RS-232 и порт CENTRONIX).

Изготавливаем и поставляем

станки для изготовления шлако- и керамзитобетонных стеновых блоков размером 400×200×200 мм.

Высылаем документацию:

комплект рабочей документации для изготовления малогабаритного станка для изготовления шлако- и керамзитобетонных блоков;

книги и брошюры по программному и аппаратному обеспечению ПЭВМ «Синклер ZX Спектрум»;

справочник адресов и телефонов зарубежных фирм, аккредитованных в Москве.

Предлагаем

быстро продать или купить стройматериалы, товары, сырье (только организациям).

Предоставляем

брокерские услуги на московских биржах.

Оказываем

помощь организациям и предприятиям в заключении контрактов с аккредитованными в Москве иностранными фирмами и СП на поставку за рубли партий малогабаритной сельхозтехни-

ки, импортного промышленно-технологического оборудования, телерадиоаппаратуры, электротоваров, одежды, обуви, бытовых товаров. Цены — минимальные (в 1,5—2 раза ниже биржевых).

Приобретаем СКВ за рубли (только у организаций).

Оказываем содействие в приобретении СКВ под контракты.

Для получения бесплатных списков контрактных товаров и перечня услуг фирмы «Восток Лтд» и кооператива «Электрон» направляйте заявки по адресу:

103489, Москва, Зеленоград, корпус 705, «Восток Лтд», «Электрон».

Проезд: от метро «Речной вокзал» автобус № 400 в г. Зеленоград до остановки «Кинотеатр «Эра», далее автобусами № 1, 2, 6, 7, 10 до остановки «Поликлиника № 65». Вход с торца корпуса 705, кооператив «Электрон» и фирма «Восток Лтд». Время работы: с 11 до 18 ч, кроме субботы и воскресенья.

**Всем руководителям учебных заведений,
использующим учебные терминальные классы
на основе IBM-совместимых ЭВМ**

**Научно-техническое предприятие «Альтернатива» при НИИ
ИВТ АПН СССР изучает спрос и принимает предваритель-
ные заявки на программный пакет Альтернатива-IBM**

Пакет предназначен:

для обеспечения эффективной работы учебных классов;

для разработки новых учебных программных пакетов;

для обучения основам информатики.

В состав пакета планируется включить следующие программы:

1. Сетевой монитор, обеспечивающий эффективную пересылку программ и файлов по локальной сети.
2. Монитор — оболочка пакета, обеспечивающая интегрированное использование программ пакета.
3. Систему управления базами данных.
4. Текстовый процессор.
5. Графический редактор.
6. Электронные таблицы.
7. Клавиатурные тренажеры.
8. Авторскую систему для построения учебных курсов.
9. Учебный Турбо-Паскаль.

Планируется обеспечить:

полную лицензионную чистоту пакета;

конкурентоспособность пакета по техническому уровню с иными известными разработками отечественных и зарубежных фирм;

комплектование пакета подробной технической и методической документацией;

возможность работы пакета на всех типах IBM-совместимых ЭВМ;

возможность его модульной настройки под возможности конкретной ЭВМ (IBM PC/XT/AT/PS-2 и их аналоги);

самые низкие цены (порядка 2000 рублей за копию).

Заявки, заказы, пожелания и предложения просим направлять по адресу:

656049, г. Барнаул, а/я 3475, «Альтернатива».

Е. БАКМАН, И. ДРЕМОВА

Черниговский педагогический институт

Введение в микро-Пролог

Точные определения

До сих пор мы использовали понятия *имя*, *предикат*, *предложение* и другие без точного их определения. Пришло время исправить положение. Одновременно с определением новых понятий мы будем их «объяснять» Прологу, заносим определения в базу знаний.

Самыми элементарными кирпичиками, из которых строится предложение, являются термы. Терм — это или имя, или переменная, или число, или список термов.

Имя состоит из букв (за исключением отдельных букв X, Y, Z), цифр и знаков. В именах запрещается использование пробела. Если нужно записать несколько слов как одно имя, то между ними ставится черточка, например: ЖИЛИ—БЫЛИ или ДЕД—ПО—ОТЦУ. Имена используются для обозначения объектов и отношений. Они являются нечисловыми константами и не должны состоять из одних цифр.

Переменные начинаются с букв X, Y, Z, за которыми может следовать целое положительное число, например: X1, Y2, Z10. Переменные могут обозначать любую конструкцию Пролога: имя, число, предикат, предложение. Такой способ обозначения переменных не позволяет делать их имена содержательными, как в алгоритмическом языке или Паскале.

Пролог допускает и действия над числовыми данными. Числа обозначаются так же, как принято в языках программирования. Они обозначают числовые объекты, поэтому могут входить в отношения в качестве одного из аргументов, например: ПОЛУЧАЕТ—ЗАРПЛАТУ РУБЛЕВ 150, что означает: Рублев получает зарплату 150 рублей. В микро-Прологе допускаются только целые числа.

Список термов состоит из круглых скобок, внутри которых записаны термы. В качестве разделителей между термами ставятся пробелы. Список термов сам является термом,

поэтому среди его элементов в свою очередь могут быть списки термов. Например, (СУММА 2 3 (4 5)). Данный список состоит из четырех элементов: СУММА — имя, 2 и 3 — числа, (4 5) — список термов.

Всего мы ввели четыре исходных понятия, которые известны и Прологу благодаря встроенным в него одноместным предикатам: VAR — переменная (от variable), NUM — число (от number), LST — список (от list — список) и AT — имя (от atom — неделимый). 37

Ниже приведены примеры, когда встроенные предикаты принимают значения «верно» и «неверно».

«Верно»:

```
VAR X1 . NUM 15
AT A . LST (3)
NUM -4, AT 10K
VAR Z . LST (3 X)
```

«Неверно»:

```
VAR A . NUM A
AT X1 . LST 15
VAR 1X . NUM P
LST X . AT B
```

Вы можете и сами проверить истинность предикатов с помощью вопросов. Например:

```
.. VAR X
..
```

Такой ответ (только приглашение) означает «верно» — X является переменной (внимание: если предыдущими командами переменной X присваивается значение, то VAR X будет «неверно»).

```
.. VAR A
?
..
```

В ответ на этот вопрос был напечатан вопросительный знак, что означает «неверно» — A не является переменной (это имя*).

Проверку истинности можно делать несколькими способами.

1. Если отношение имеет один аргумент, то проще всего набрать имя отношения и аргумент без всяких скобок, как это сделано выше.

2. Если аргументов несколько или ни одного, то приходится для обозначения конца аргументов выражение брать в скобки:

<Имя отношения> <arg 1>...<arg n>

3. Если же проверяется конъюнкция предикатов, то каждый предикат заключается в скобки, все они вместе заключаются еще в одни скобки и спереди ставится вопросительный знак. Например:

_.? ((ДОЧЬ ОЛЬГА X) (PP X))

Последующие понятия являются производными, и мы их определим с помощью правил.

Предикат — это список термов, первый из которых — имя. Это имя является именем предиката и называется еще предикатным символом. Например, выражение (СЕСТРА X КОЛЯ) является предикатом, так как это список термов и в качестве первого терма стоит предикатный символ СЕСТРА. В естественном языке предикату соответствует отношение «быть сестрой» — X является сестрой Коли.

Еще один встроенный предикат — (LESS X Y). Этот двухместный предикат определяет отношение «быть меньше, чем». Он принимает значение истина, если $X < Y$.

Обратите внимание, что в математике отношения записываются с помощью специальных знаков $<$, $=$ и т. д., а в Прологе (как и в естественном языке) все отношения обозначены именами. Кроме того, в математике знаки отношений ставятся между операндами (инфиксивный способ), а в Прологе имя отношения стоит перед операндами (префиксный способ). Этот вариант более универсален; где, например, поставить знак операции в случае трехместного отношения?

Объясним Прологу, что такое предикат:

_.ADDCL ((ПРЕДИКАТ (X|Y)) (LST (X|Y)) (AT X))

Это означает: (X|Y) является предикатом, если (X|Y) список (LST) и X — имя. Здесь мы применили вертикальную черту для того, чтобы выделить X — первый терм аргумента. Через |Y обозначен остаток списка, он может содержать любое число элементов, в частности быть пустым.

Такая же вертикальная черта была использована при определении отношения КАКИЕ. С ее помощью был отделен X — первый элемент списка, который затем печатался с помощью команды PP X. А через |Y обозна-

чался остаток списка, который должен был проверяться на истинность командой

? Y

Проверим теперь, как работает введенное в базу знаний отношение «быть предикатом».

_.ПРЕДИКАТ(3 5)

? Y

.-

Неверно, так как первый элемент списка число 3, а должно быть имя.

_.ПРЕДИКАТ(ГАЗ-ВОДА 3)

.-

Теперь верно. (ГАЗ — ВОДА 3) — это предикат.

Предложения, правила и квантор всеобщности

Предложение — это список предикатов. Раз это список, входящие в него предикаты заключаются в круглые скобки, а так как каждый предикат в свою очередь является списком термов, то предложение всегда начинается как минимум с пары открывающихся скобок. Пример предложения: ((СНЕСЛА КУРОЧКА ЯИЧКО)). Это предложение состоит из одного предиката (СНЕСЛА КУРОЧКА ЯИЧКО), взятого в скобки, чтобы превратить его в список предикатов. Такое предложение мы называли фактом.

Еще пример:

((ДЕД X Y) (РОДИТЕЛЬ Z Y)) (ОТЕЦ X Z)

Такое предложение мы называли правилом. Оно означает: X является дедом Y, если найдется Z — родитель Y, причем X должен быть отцом Z.

Объясним термин *предложение* Прологу:

_.ADDCL ((ПРЕДЛОЖЕНИЕ (X)) (ПРЕДИКАТ X))

_.ADDCL ((ПРЕДЛОЖЕНИЕ (X|Y)) (ПРЕДИКАТ X) (ПРЕДЛОЖЕНИЕ Y))

Первое введенное правило означает, что список из одного предиката является предложением. А второе правило рекурсивное: предложением является список, в котором первый элемент — предикат, а остаток списка — предложение.

Руководствуясь вторым правилом, Пролог будет выделять предикаты по одному из начала списка до тех пор, пока не останется в списке один предикат. Тогда срабатывает первое определение предложения и результат будет «верно». Если же попадетесь хоть один

не предикат, то проверка прекратится с результатом «неверно». Например:

.. ПРЕДЛОЖЕНИЕ (ПЯТЕРЕНКИ ШЕСТЕРЕНЧАТЫЕ)

Нам еще нужно уточнить, как работает правило. Рассмотрим это на примере определения отношения ДЕД, приведенного выше.

Первый предикат (ДЕД X Y) является заключением. Любое правило выполняется для всех значений переменных, которые входят в заключение. Заданное отношение ДЕД является правилом, поэтому подразумевается, что все значения переменных X и Y, удовлетворяющие условиям (РОДИТЕЛЬ Z Y) и (ОТЕЦ X Z), находятся в отношении ДЕД.

Как было сказано выше, условие (или цель) представляет собой конъюнкцию отдельных подцелей — предикатов, следующих за заключением. Знак конъюнкции не пишется, как, впрочем, часто делается и в математической литературе.

Для проверки истинности заключения при некоторых X и Y должна быть доказана истинность всех подцелей. Следует обратить внимание на то, что в подцелях встречается не входящая в заключение переменная Z. Это нужно понимать так, что некоторое Z удовлетворяет все подцели. Употребляя слово «некоторый» в обычном языке, мы имеем в виду «найдется хотя бы один» или «существует хотя бы один». Напомним, что выражение «хотя бы один» означает один или более. Если найти подходящее Z не удастся, то считается, что заключение ложно.

Как же работает исполнительная система Пролога? Для каждого запроса она ищет в базе знаний правило или факт, в котором имя предиката совпадает с именем, стоящим в запросе. Если истинность предиката удается доказать (все подцели истинны), то результатом будет «верно» и процесс прекращается. В противном случае ищется следующее правило с нужным предикатным символом. Если истинность не удается доказать, то ответом будет «неверно».

Прологовское «неверно» не всегда означает «ложь», просто не нашлось в базе знаний подтверждения запроса. Возможно, это произошло из-за отсутствия сведений. Правда, для встроенных предикатов над числами «неверно» значит «ложно»; пример — (LESS 3 2).

Рассказывая о работе исполнительной системы Пролога, мы ни разу не сказали, что он делает с операциями. Для стандартизации записи предложений все операции считаются также отношениями и имеют вид предикатов, например в команде печати имя PP считается предикатным символом, за которым

следует любое число аргументов до закрывающей скобки. Значением такого предиката всегда считается истина, за исключением случаев сбоя оборудования.

Особое положение занимает встроенное отношение FAIL (неудача), которое никогда не удовлетворяется. Оно приблизительно соответствует константе «ложно», но, как уже говорилось, в Прологе «ложно» — редкое явление.

Теперь мы уже готовы рассмотреть данное во втором параграфе определение предиката КАКИЕ. Правило

((КАКИЕ (X1Y)) (? Y) (PP X) FAIL)

означает, что значение «верно» получится, если верно Y, успешно напечатается X и верно FAIL. Но FAIL не бывает верно, это заставляет Пролог искать другие значения X и Y и для каждого верного Y печатать X.

Таким образом, FAIL заставляет перебирать и печатать все нужные нам X, а не останавливаться на первом значении X, удовлетворяющем Y. Так например, при запросе

- КАКИЕ (X (РОДИТЕЛЬ X ОЛЬГА))

происходит сопоставление запроса с заключением приведенного выше правила. В результате Y принимает значение

..(РОДИТЕЛЬ X ОЛЬГА)

Далее идет проверка истинности подцелей, среди которых первой в правиле стоит (? Y). Если Y верно, то следующим предикатом является PP X, который печатает X, т. е. имя одного из родителей Ольги. Но затем FAIL заставляет искать другого родителя, так как последняя подцель оказывается неверной.

Обратите внимание, что FAIL можно писать без скобок.

Арифметические отношения и рекурсия

В Прологе имеется трехместный предикат SUM X Y Z. Если все три его аргумента имеют конкретные значения, например SUM 2 3 5, то это действительно предикат и он принимает значение «истина» в случае, если $Z = X + Y$.

Если же одна из переменных не имеет конкретного значения, то этот предикат представляет собой операцию сложения или вычитания. Так, SUM 3—2 Z присвоит Z значение $3 + (-2) = 1$. А SUM Y 5 9 присвоит Y значение $9 - 5 = 4$. Причем значение самого предиката всегда «истина» во всех случаях использования его в качестве операции.

Использование отношения SUM с двумя

неизвестными аргументами недопустимо, так как тогда получаем бесконечное множество решений.

Умножение и деление вводятся встроенным предикатом `TIMES X Y Z` аналогично `SUM`. Например, `TIMES 3 5 15` истинно; `TIMES 4 X 24` присвоит `X` значение 6 и будет истинным; `TIMES 6 2 Y` означает $Y = 6 \times 2$.

Если деление не выполняется нацело, то в частном отбрасывается дробная часть. Так, `TIMES X 3 8` присвоит `X` значение 2.

Похож на `SUM` и `TIMES` двухместный предикат `EQ X Y` (равенство). Так, `EQ 2 2` истинно, а `EQ X 3` присвоит `X` значение 3, если `X` еще не было определено. Если же `X` имело значение 5, то оно не изменится, а результат будет «ложно».

Пусть нам нужно вычислить $35 \times 3 - 18$. Это можно сделать так:

```
-.?((TIMES 35 3 X) (SUM 18 Y X)
    (PP Y))
```

87

Сначала переменная `X` получает значение 35×3 . Далее она уже имеет значение, поэтому вычисляется `Y` как разность $X - 18$, но $X = 35 \times 3$, поэтому $Y = 35 \times 3 - 18$, что и печатается командой `PP Y`.

Можно выполнить и более сложные расчеты. Например, $F(X) = (2X - 3)(X + 5)$ при $X = 6$, но возможны и другие значения `X`. В этом случае выражение лучше оформить в виде одноместного предиката:

```
-.ADDCL((F-OT X) (TIMES 2 X Y)
        (SUM 3 Z Y) (SUM X 5 Z1)
        (TIMES Z Z1 Z2) (PP Z2))
```

```
-.F-OT 6
```

99

```
-.F-OT -1
```

-20

--

Здесь использованы переменные $Y = 2X$, $Z = 2X - 3$, $Z1 = X + 5$, результат `Z2`. Конечно, запись выражения намного сложнее, чем в процедурных языках программирования, но достаточно определить его один раз, и можно будет обращаться к нему по имени предиката.

Нам понадобится еще команда ввода `R X`, которая также считается предикатом с именем `R`. При выполнении этой команды печатается точка (признак ожидания ввода), и компьютер останавливается. Следует набрать требуемое выражение и нажать `ВВОД`. При этом переменной `X` будет присвоено введенное выражение. Если оно окажется термом, то значение предиката `R` — «истинно», иначе «ложно».

Используем `R` для ввода имени пользователя.

```
-.ADDCL((ВВОД-ИМ-ПОЛЬЗ)
        (P КАК ВАС ЗОВУТ ?) (R X)
        (KILL ПОЛЬЗОВАТЕЛЬ)
        (ADDCL((ПОЛЬЗОВАТЕЛЬ X))))
```

При использовании получим следующую картину:

```
-. (ВВОД-ИМ-ПОЛЬЗ)
КАК ВАС ЗОВУТ ? Петя
```

После ввода «Петя» командой `(KILL ПОЛЬЗОВАТЕЛЬ)` в базе знаний уничтожаются сведения-факты об именах предыдущих пользователей и командой

```
(ADDCL((ПОЛЬЗОВАТЕЛЬ X)))
```

в базу знаний добавляется факт

```
(ПОЛЬЗОВАТЕЛЬ Петя).
```

Теперь можно использовать имя пользователя для диалога. Простейший пример:

```
-.ADDCL((ПРИВЕТ) (PP Я - ПРОЛОГ)
        (ВВОД-ИМ-ПОЛЬЗ) (ПОЛЬЗОВАТЕЛЬ Y)
        (P ЗДРАВСТВУЙТЕ, Y.)
        (PP ПРИЯТНО С ВАМИ ПОЗНАКОМИТЬСЯ))
```

```
-. (ПРИВЕТ)
```

Я - ПРОЛОГ

КАК ВАС ЗОВУТ ? Ира

ЗДРАВСТВУЙТЕ, Ира. ПРИЯТНО С ВАМИ

ПОЗНАКОМИТЬСЯ

Рассмотрим традиционный алгоритм вычисления максимума двух чисел.

```
-.ADDCL((MAX X Y) (IF (LESS X Y)
        ((PP MAX= Y)) ((PP MAX= X)) ) )
```

```
-. (MAX 3 15)
```

MAX=15

```
-. (MAX 3 -5)
```

MAX=3

--

Здесь использован трехместный предикат `IF`, который аналогичен условному оператору `IF <условие> THEN <команда> ELSE <команда>`. В данном примере предикат `IF` эквивалентен оператору

```
IF X<Y THEN печать Y ELSE печать X
```

Таким образом, условный оператор имеет вид:

```
(IF (УСЛОВИЕ) ((КОМАНДА 1))
    ((КОМАНДА 2)))
```

Условие заключается в скобки, а команды в двойные скобки. `УСЛОВИЕ` и `КОМАНДЫ` представляют собой предикаты или конъюнкцию предикатов. Подразумевается, что `((КОМАНДА 1))`, следующая за `(УСЛОВИЕ)`, будет выполнена, если `(УСЛОВИЕ)`

верно, в противном случае будет выполнена ((КОМАНДА 2)).

Можно вводить X и Y с клавиатуры:

```
_.ADDCL (( MAX ) ( P Введите
                два числа ) ( R X ) ( R Y )
                ( IF ( LESS X Y )
                ( EQ Z Y ) ) ( EQ Z X ) )
                ( PP MAX= Z )
_. (MAX)
Введите два числа. 5 8
MAX=8
```

Пока никаких существенных отличий от процедурных языков нет. Но давайте определим предикат «сумма элементов списка». В алгоритмическом языке это задача суммирования элементов таблицы. Определение будет рекурсивным: если список пустой, то сумма равна нулю, иначе сумма списка равна первому элементу плюс сумма оставшегося списка.

Вспользуемся вспомогательным отношением СУМ с тремя аргументами: первый — сумма просмотренной части списка, второй — переменная-результат, третий — остаток списка, его непросмотренная часть. Если из остатка списка взять первый элемент и прибавить к сумме просмотренной части, то мы снова получим СУМ, но уже для списка, содержащего на один элемент меньше.

```
_.ADDCL ((СУМ X Y (Z:Z1)) (СУМ X Z Y1)
                (СУМ Y1 Y:Z1))
```

Здесь $Y1 = X + Z$.

Когда весь список закончится, то останется только два аргумента, которые должны совпадать. Поэтому добавим

```
_.ADDCL ((СУМ X X ( ) ))
```

Знак () означает, что все элементы списка исчерпаны, список пуст.

Тогда сумму списка определим с помощью

```
_.ADDCL ((СУММА-СПИСКА X:Y) (СУМ 0 X:Y))
```

Здесь X — сумма элементов списка, а Y — список.

Проверка:

```
_.?((СУММА-СПИСКА X (3 6 -2)) (PP X))
7
_.
```

Можно сделать так, чтобы список вводился по приглашению.

```
_.ADDCL ((СУМ-ВВЕД-СП) (P Введите
                список чисел)
                (R Y) (СУММА-СПИСКА Z:Y) (PP S= Z))
```

Посмотрим его работу:

```
_. (СУМ-ВВЕД-СП)
Введите список
чисел. ( 9 15 26 -4 )
S=46
```

Заметим, что при введении списка его элементы следует заключать в круглые скобки.

Окончание следует.

41

М. КОВТУН

Программирование в системе Express-Pascal

В этом цикле статей будет рассказано о языке программирования Паскаль и его использовании. Изложение будет вестись на примере системы программирования Express Pascal, реализованной на ПЭВМ «Корвет».

Введение

Чтобы заставить компьютер что-нибудь сделать, ему нужно дать программу выполнения этого действия. А программу нужно

сформулировать на каком-то языке. Языки, на которых мы пишем программы,— это и есть языки программирования.

Паскаль — один из них. Он появился в конце 60-х гг. и сейчас входит в число самых популярных. Изначально Паскаль задумывался как учебный язык. Из этой цели вытекает несколько требований: во-первых, учебный язык программирования должен быть простым; во-вторых, в нем должны быть

представлены все основные концепции построения языков программирования; в-третьих, работа на этом языке должна приучать к правильному стилю программирования. Паскаль блестяще соответствует всем этим требованиям. Более того, оказалось, что Паскаль годится и для профессионального программирования, так что ныне многие широко используемые программы пишутся на Паскале.

Этот язык действительно прост: его полное формальное описание с комментариями умещается на 30 страницах (для сравнения: подобное описание языка Алгол-68 — книга объемом около 500 страниц). Но простота достигнута не за счет отказа от каких-то возможностей, а благодаря очень удачной компоновке языка и использованию простых и ясных конструкций. Дополнительные элементы (расширения) языка, реализованные в системе Express Pascal (в качестве образца для которой была взята система Turbo Pascal), делают его пригодным для написания практически любых программ, в том числе и так называемых системных, предъявляющих очень высокие требования к языку программирования (например, программа «Дисковый редактор» для «Корвета» написана на Express Pascal).

Важной отличительной чертой Паскаля является высокая надежность программирования. Что это такое, лучше всего показывает одна история из программистского фольклора. Рассказывают, что серьезные неприятности при запуске одного из американских космических кораблей возникли из-за маленькой ошибки в программе, написанной на Фортране. В одной из строк вместо `DO 5 I=1,10` было написано `DO 5 I=1.10`

(как быстро вы заметили, чем отличаются эти строки?). Эта почти незаметная глазу описка — точка вместо запятой катастрофически исказила смысл программы. Знающие Фортран легко поймут, в чем дело: компилятор распознал строку не как заголовок оператора цикла (чем она должна была быть), а как оператор присваивания: переменной `DO5I` присваивается значение `1.10`. Пробелы в Фортране игнорируются, поэтому запись `DO 5 I` полностью эквивалентна записи `DO5I`; переменные объявлять заранее не нужно, поэтому компилятор просто завел новенькую переменную. А в результате программа вычислила совсем не то, что было нужно.

Синтаксис Паскаля (правила записи программ) устроен так, что подобные ошибки практически исключаются. Да и многие иные ошибки, которые причинили бы немало забот

при работе с другими языками программирования, вы легко обнаружите с помощью Паскаль-компилятора.

Скажем несколько слов о способах реализации языков программирования. Изначально компьютер понимает только язык машинных команд — последовательности нулей и единиц. Чтобы научить его более сложному (для компьютера, но более простому для человека!) языку, нужно снабдить его соответствующей программой.

Программы, обучающие компьютер новым языкам, бывают двух типов: интерпретаторы и компиляторы. Интерпретатор заставляет компьютер исполнить программу на языке программирования в том виде, в каком она была написана человеком: ЭВМ берет очередной оператор программы, анализирует его и выполняет предписанные оператором действия. Если приходится повторно выполнять какой-либо оператор, то анализ его выполняется заново. Всякий раз, когда вам нужно выполнить программу, вы должны запускать интерпретатор.

Компилятор переводит программу с языка программирования на язык машинных команд, и компьютер исполняет программу в «родных» для него машинных кодах. Само исполнение происходит, естественно, без всякого участия компилятора: он уже сделал свое дело, осуществив перевод.

Оценим достоинства и недостатки интерпретаторов и компиляторов.

Достоинства интерпретатора. Программу можно выполнить, как только она написана (не тратя времени на компиляцию). На ПЭВМ интерпретатор всегда объединен с редактором текстов, поэтому, если в программе обнаружится ошибка, ее можно быстро исправить и немедленно запустить программу снова.

Недостатки интерпретатора. При каждом запуске программы в памяти компьютера кроме ее текста должен находиться интерпретатор; это сильно ограничивает максимальный размер программы. При каждом исполнении каждого оператора интерпретатор выполняет его анализ заново; на это уходит много времени, в результате скорость выполнения программы уменьшается приблизительно на порядок по сравнению со скоростью выполнения скомпилированной программы. По этим причинам большую программу, которую нужно будет постоянно использовать, создавать с помощью интерпретатора неэффективно.

Достоинства компилятора. Он порождает компактную и быстро работающую программу.

Недостатки компилятора. Процесс создания программы занимает много времени.

При использовании традиционно построенного компилятора приходится выполнять такую последовательность действий: подготовить текст программы с помощью редактора текстов; скомпилировать программу; отредактировать связи (присоединить к вашей программе написанные другими программистами кусочки, выполняющие стандартные действия); запустить полученную программу в машинных кодах. Если в ней обнаруживается ошибка, приходится возвращаться к исходному тексту, исправлять ошибку и повторять весь цикл заново. На таком компьютере, как «Корвет», исправление одной ошибки в самой маленькой программе занимает около 5 мин (а в большой программе — еще дольше).

Если вы работали на «Корвете» с Бейсик-интерпретатором и компилятором с Паскаля МТ+, вы на своем опыте постигли все сказанное.

По-видимому, Борланду первому пришла в голову мысль о том, как соединить достоинства компилятора и интерпретатора; эту мысль он реализовал в своей системе Turbo Pascal. (После того как идея реализована, она кажется очевидной. Но не так просто до нее додуматься. Первой моей реакцией, когда я услышал о системе Turbo Pascal, было: «Это невозможно!»)

Идея крайне проста: создать интегрированную систему, включающую редактор текстов и компилятор. Вы вводите текст программы, пользуетесь редактором; после того как ввод закончен, запускаете компилятор (он уже находится в памяти, и времени на его загрузку не требуется!); компилятор размещает рабочую программу в памяти, и ее можно тут же запустить. Если компилятор обнаружит ошибку в программе, он запускает редактор и устанавливает курсор в том месте текста, где находится ошибка. При работе с маленькой программой время, затрачиваемое на цикл «исправление ошибки — компиляция — запуск», составляет всего несколько секунд — потери по сравнению с интерпретатором практически незаметные. Скомпилированную программу можно записать на диск и выполнять впоследствии без участия интегрированной системы.

Express Pascal является именно такой интегрированной системой. Входной язык системы является расширением языка Turbo Pascal V. 4.0 для компьютеров типа IBM PC. Если в программе с IBM PC не использовались машиннозависимые черты языка, то ее можно без всяких изменений скомпилировать и выполнить на «Корвете».

Если вы спросите, каким языком лучше пользоваться при работе на «Корвете» —

Бейсиком или Паскалем — и какому языку лучше обучать школьников, наш ответ будет однозначным: конечно, лучше Паскаль. Работать с системой Express Pascal ничуть не сложнее, чем с Бейсик-интерпретатором, а возможностей она предоставляет больше; человек, научившийся писать программы на Бейсике, приобретает и навыки, которые будут мешать ему писать серьезные программы, в то же время навыки, приобретенные при работе с Паскалем, обеспечивают хороший стиль программирования и позволяют легко осваивать новые языки.

В этом цикле статей мы постараемся научить вас писать программы на Паскале и работать с системой Express Pascal. При описании работы с компьютером мы предполагаем, что у вас есть Express Pascal и документация к нему. В документации подробно описано, как работать с системой, и приведено подробное формальное описание языка. Поэтому в «ИНФО» мы будем обращать внимание в первую очередь не на то, как выполнить какое-либо действие (это можно узнать из документации), а на то, когда, зачем и почему это действие можно выполнить. По сути этот цикл статей является учебником, дополняющим формальную документацию.

Простейшие элементы языка Паскаль

Давайте сначала напишем и выполним очень простую программку:

```
var x : integer;
begin
  Write('Введите число: '); Readln(x);
  x := x*x;
  Writeln('Квадрат введенного
          числа = ', x);
end.
```

Она вводит целое число, вычисляет его квадрат и выводит его на экран.

Обратите внимание на то, как размещен текст программы. Чтобы программу было легко читать и изменять, текст следует располагать некоторым регулярным образом. Сейчас мы сформулируем несколько первых правил, которых рекомендуется придерживаться.

* В каждой строке должен стоять только один оператор. Несколько операторов можно размещать в строке только в том случае, если они образуют неразрывную последовательность действий, как, например, в третьей строке примера: находящиеся в ней операторы выполняют выдачу вопроса и получение ответа.

* Слова **begin** и **end** всегда появляются парами. Парные слова **begin** и **end** должны либо находиться в одной и той же строке, либо начинаться с одной и той же позиции;

в последнем случае в строках, где стоят **begin** и **end**, не должно быть ничего другого.

* Если одна конструкция (**begin** — **end**, цикла и т. п.) вложена в другую, то строки вложенной конструкции должны начинаться правее строк объемлющей конструкции. В примере строки, расположенные между **begin** и **end**, сдвинуты на две позиции вправо.

Далее по мере описания новых конструкций мы будем формулировать правила их записи.

Попробуйте скомпилировать и выполнить эту программку. Для этого:

выйдите в основное меню системы Express Pascal (если вы находитесь в редакторе, нажмите для этого **Esc**);

нажмите клавишу **C**; программа будет скомпилирована, в нижней части экрана появится сообщение о завершении компиляции; нажмите **Esc**, система вернется в основное меню;

нажмите клавишу **P**; начнется выполнение программы;

в верхней строке экрана появится «Введите число:»; введите в ответ какое-нибудь число (например, 25) и нажмите **VK**;

во второй строке экрана появится «Квадрат введенного числа = 625», а в левом нижнем углу — «Press Esc». Ваша программа закончила свою работу, и теперь система ожидает, пока вы ознакомитесь с результатами ее трудов. Нажав **Esc**, вы вернетесь в основное меню.

Теперь займемся программой. Почему эти

строки делают то, что они делают, — вводят число и выводят его квадрат? Почему мы написали их так, а не иначе?

Наша программа работает с числом. В период работы это число должно где-то храниться. Место, где программа хранит число, называется *переменной*. Переменную удобно понимать как ящик, снабженный этикеткой — *именем переменной*. По этому имени мы можем обращаться к переменной. Ящички-переменные могут иметь разные формы и размеры, определяющие, что можно положить в ящик. Форма и размер ящичка — это *тип переменной*.

В Паскале каждая переменная, используемая в программе, должна быть *описана*. Для описания переменной и используется первая строка нашей программы:

```
var x : integer;
```

Слово **var** служит признаком того, что дальше будет идти описание переменных. Каждый раз, когда описывается переменная (или группа переменных), нужно употребить это слово. Слово **x** — имя переменной. Имена переменных мы придумываем сами. Они могут состоять из латинских букв, цифр и знаков подчеркивания, но начинаться должны с буквы. В системе Express Pascal имя переменной может быть почти любой длины (единственное ограничение — оно должно целиком помещаться в одной строке, а ее максимальная длина — 127 литер).

Продолжение следует.

Я. ФУФЫГИН

Новый текстовый редактор для «Агата»

Общеизвестно, что в большинстве случаев персональные компьютеры используются либо как пишущие машинки, либо, увы, как игровые автоматы. На компьютерную обработку текстов нацелен целый класс программ — текстовые редакторы, наличие которых в базовом программном обеспечении давно стало стандартом для любой ПЭВМ. Для БК-0010, например, имеется уже около 30 редакторов, и число их продолжает увеличиваться.

Важность этого класса программ и необходимость приобретения навыков работы с ними еще в средней школе очевидны, и здесь мы намерены подробнее рассмотреть,

какие возможности предоставляет для этого вторая по популярности в нашем образовании (после БК-0010) персональная ЭВМ «Агат».

«Агат» никак нельзя назвать технически ущербной машиной. Он имеет немало преимуществ перед другими отечественными ПЭВМ, поставляемыми в школы. Наличие дисководов на каждой машине, модульный принцип построения компьютера, большое ОЗУ (96—768 К), достаточно быстрый процессор 6502, такой же, как в популярнейшей машине Apple II, возможность подключить принтер к любой машине в классе — адаптеры стоят на всех — приближают этот

компьютер к «настоящим» машинам.

Однако для «Агата» сложилась абсолютно ненормальная ситуация в плане обеспечения его программами, и особенно в классе текстовых редакторов.

Первый из них — текстовый редактор системы «Школьника». Он изначально был ориентирован на работу с текстами программ, а не документов и содержит лишь тот минимум функций, который необходим для работы с программным текстом, в то время как современные системы подготовки текстов должны иметь значительно более высокий уровень сервиса — средства оформления и форматирования текста, импорт графики, управление принтером и многое другое. Этот же редактор не содержит в себе даже элементарных функций форматирования текста и управления печатью. Следует признать, что он мало приспособлен для работы, отличной от написания программ.

Следующий кандидат — система «Агат-автор». Это уже более профессиональный инструмент, который при создании ориентировали именно на подготовку не столько программ, сколько документов общего содержания. Он содержит основные средства форматирования текста, в нем можно исполнять вполне качественно оформленные документы. Но эта система имеет ряд серьезных недостатков, прежде всего не совместимый с другими системами собственный формат записи. Кроме того, «Агат-автор» не относится к нормальным программным продуктам, работающим в среде Apple DOS. Он сам загружается вместо системы, и для выхода из него необходимо перезапускать машину. «Агат-автор» не способен также работать с расширенным знакомателем «Агата», который ставится на последние выпуски этой ПЭВМ. Работать с ним можно только в режиме 32 символа в строке. Для управления шрифтами и форматированием документа в «Агат-авторе» используется система двухбуквенных команд, вставляемых в текст, что мешает восприятию текста на экране и затрудняет работу с редактором.

Другие текстовые редакторы в силу каких-то причин не получили широкого распространения.

Острая необходимость в добротном текстовом редакторе для «Агата» побудила нас приступить к разработке собственного программного продукта такого типа. Он был написан за четыре месяца на языке ассемблера и немедленно запущен в эксплуатацию. При разработке были учтены следующие требования:

редактор должен не подменять собой DOS, а являться обычной программой в машинном коде, не нарушающей работу DOSa.

редактор должен обслуживать одновременно дисководы как 140, так и 800 К;

редактор должен обеспечивать печать графических иллюстраций к тексту;

редактор должен иметь простую и ясную систему управления процессом создания и редактирования текста — современную систему меню и подсказки;

редактор должен обеспечивать работу с текстами неограниченной длины, не уступающими в ОЗУ;

редактор должен позволять максимально полно использовать возможности печатающего устройства и легко управлять ими.

Созданная система достаточно полно отвечает этим требованиям. Она проста в освоении и по идеологии близка к сходным системам для IBM PC.

Опишем вкратце работу с ней.

После загрузки с диска основное меню предлагает следующие варианты действий: создать новый текст, перейти к работе с файлами, перейти к распечатке, перейти к редактированию или вводу текста, закончить работу с системой.

При выборе какой-то из позиций происходит переход либо к меню второго уровня (их два — для работы с файлами и для управления режимами печати), либо сразу в нужный режим.

Самый главный режим — редактирование текста. От удобства и развитости функций редактирования во многом зависит общий уровень комфорта при работе с редактором. Что же может эта система?

Для управления редактированием использованы функциональные клавиши «Агата». Их всего 15 штук, поэтому некоторые команды отдаются с префиксом, в качестве которого использована клавиша F3. Всего в системе 28 функций редактирования текста, в том числе:

ввод текста при редактировании в режиме вставки или замены;

вставка в текст команд переключения шрифтов принтера;

вставка строк;

стирание фрагмента из нескольких строк с запоминанием в буфере;

вставка в текст запомненных в буфере строк;

удаление строк без запоминания;

задание модели контекстного поиска;

поиск модели в тексте;

склейка строк;

транслитерация (превращение строчных букв в прописные и наоборот);

центровка строк;

рассечение строки в позиции курсора;
перемещение по тексту (на 12 строк вверх или вниз, в конец текста, в начало текста, в начало и конец текущей строки);
выравнивание абзацев в тексте;

сжатие пробелов во всем тексте или только в текущей строке;

просмотр текста в цветном режиме 32×32 с обозначением прописных и строчных букв (для «Агата-7»);

запоминание и исполнение до четырех макрокоманд (до 254 команд в каждой);
переформатирование фрагмента текста под необходимую длину строки;

выравнивание текста по левому или правому краю;

выдача подсказки о назначении клавиш.
Режим работы с файлами включает в себя следующие возможности:

выбор рабочего дисководов;

просмотр каталога диска;

запись текста в текстовый файл;

считывание текста;

просмотр текстового файла без загрузки в память;

объединение текстовых файлов в памяти;
временный выход в DOS без потери текста в памяти;

запись на диск фрагментов текста из памяти.

Необходимо остановиться подробнее на такой возможности, как временный выход в DOS. Такая функция довольно обычна для машин типа IBM PC, но в программном обеспечении «Агата» ранее не встречалась. Осуществляется она весьма просто: после выбора соответствующей позиции в меню пользователь переходит в обычный диалог DOS/BASIC. Он может исполнять любые команды (даже INIT и FP), выполнять расчеты в калькуляторном режиме Бейсика и даже запускать небольшие программы на

этом языке. Для возврата в редактор достаточно в ответ на стандартное приглашение DOS нажать F3 — и мы снова в редакторе, даже текст в памяти не поврежден, если только его не подпортила запускаящаяся программа на Бейсике (чтобы этого избежать, можно поставить в Бейсик-программу строку 0 LOMEM:16383).

Имеются достаточно широкие возможности управления принтером. Все параметры задаются в специальном меню, которое появляется на экране после выбора режима печати. К ним относятся:

межстрочный интервал при печати;

левое поле;

шрифт;

режим печати (обычный или качественный);

автоматическое разбиение на страницы (есть/нет);

автоматическая нумерация страниц (есть/нет);

задание числа экземпляров печатаемого документа;

задание колонтитула.

Возможны также изменения по ходу печати для выделения отдельных мест текста жирным, расширенным или каким-то иным шрифтом.

Рекомендуемый принтер — CM 6337 или любой другой совместимый; драйвер принтера не встроен в систему, а является отдельной программой в кодах, загружаемой через специальную функцию меню, что обеспечивает возможность его легкой замены для работы с различными печатающими устройствами.

Желающие приобрести нашу систему подготовки текстов могут обращаться по адресу: 170026, Тверь, ул. Горького, 21/3, кв. 12. Фуфыгину Я. Б. Телефон в Твери: 1-38-05.

Я. ВАГРАМЕНКО, П. КОЛЫХАНОВ, Г. ЛЮБАВИНА

Банк профессиональных знаний

Банк профессиональных знаний (БПЗ) реализован как пакет информационно-справочных систем на основе баз данных иерархического типа (БДИТ). Разделы банка — базы данных — наполняются информацией из различных областей знаний, представляющих интерес в связи с проблемами повышения квалификации, переподготовки и переобучения специалистов, а также в связи

с проблемами, возникающими при переходе к новым формам экономических отношений. Базы данных предоставляют возможность быстрого и осмысленного поиска информации с помощью иерархической системы каталогов или с помощью нескольких систем ключевых слов. Информация структурирована таким образом, чтобы у пользователя в процессе работы с си-

стемой формировались представления и знания не только о тех разделах, к которым он стремится получить доступ, но и об общей структуре информации. Такой подход делает систему ценным средством с точки зрения обучения и повышения квалификации пользователя.

Информация в базе данных организована в модули, между которыми осуществляются переходы и формируются новые модули. Выделяются два типа модулей — каталоги и концевые модули. В модулях-каталогах содержатся другие модули и пользователю предлагается выбрать нижележащий модуль по его названию — строчке каталога. Система вложенных модулей-каталогов дает сведения о структуре и взаимосвязи информации. Знание этой структуры не менее ценно и информативно, чем владение информацией, содержащейся в концевых модулях. Концевые модули выдают пользователю некоторые сообщения. В простейшем случае это порция текстовой информации (текстовые модули). В качестве концевого модуля могут обрабатываться некоторые программы или базы данных нижнего уровня. К каждому модулю, кроме текстового, может прилагаться текстовая информация в виде краткой аннотации. Помимо иерархического поиска осуществляется также поиск концевых модулей по совпадению набора ключевых слов в выбранном модуле-каталоге. При этом формируется временный модуль-каталог из выбранных концевых модулей. При необходимости можно осуществить контекстный поиск во всем информационном поле, контролируемом из выбранного модуля. Однако такой поиск при больших объемах информации нецелесообразен. По ходу работы с базой данных можно получить информацию о текущем и нижележащих модулях в виде твердой копии на принтере.

В настоящий момент банк профессиональных знаний содержит пять баз данных: «Основы хозяйственного расчета», «Основные принципы конверсии», «Игровые методы обучения», «Информатика», «Трудовые ресурсы и занятость».

Характеристику разделов БПЗ можно дать на примерах, относящихся к проблемам экономики.

«Основы хозяйственного расчета».

Эта область знаний раскрывается в разрезе таких основных разделов, как показатели спроса, ресурсы и фонды, оформление коммерческих отношений, организация рентабельной работы, предъявление товара. Например, в теме «Оформление коммерческих отношений» раскрываются такие вопросы: взаимные обязательства заказчика и испол-

нителя, государственное регулирование коммерции, планово-финансовое обеспечение. Все сведения о хозрасчете представлены в виде 225 «кристаллизованных» сообщений.

«Трудовые ресурсы и занятость».

Эта чрезвычайно важная проблема, связанная с формированием рынка труда, в БПЗ представлена наукоемкими сообщениями, сгруппированными в разделы: воспроизводство трудовых ресурсов, качество трудовых ресурсов, распределение трудовых ресурсов и занятость, эффективное использование трудовых ресурсов. В частности, разъясняются демографические условия, законы регулирования спроса на труд, подготовка к труду и характеристика рабочей силы, организация и условия труда.

«Основные принципы конверсии».

Знания в этой области оказываются существенно необходимыми в современных условиях переориентации военной промышленности на рынок и гражданскую продукцию массового спроса.

Методология решения задач конверсии, а также отечественный, пока еще ограниченный опыт ее проведения наряду с зарубежным опытом, раскрыта в таких темах БПЗ: основные понятия и проблемы конверсии, мировой опыт и национальные программы, конверсия производства, конверсия вооружений и вооруженных сил. Освещены проблемы занятости освобождаемых контингентов персонала, альтернативное использование ресурсов, особенности и трудности конверсии, различные предложения по ее реализации. Вся совокупность сообщений представляет некоторое введение в эту комплексную область знаний.

В разделе «Игровые методы обучения» представлены:

общая теория игр (через схему театра, типологию деятельностных позиций в театре);

функции игр;
типология игр;
специфика организационно-деятельностных игр;
типовой сценарий игры, цели и задачи игры (на примере ОДК «Проектирование учебного процесса для резерва руководителей»).

Ведется разработка перспективных направлений и непрерывно осуществляется совершенствование и дополнение созданных баз.

Программный продукт «Банк Профессиональных Знаний» реализован для класса IBM-совместимых компьютеров и может использоваться даже в минимальной конфигурации IBM PC/AT с монитором CGA или Hercules. При необходимости разработ-

чики осуществляют адаптацию системы на отечественные персональные компьютеры типа IBM PC. Система управления «Банком Профессиональных Знаний» создана на языке СИ и отлаживается в среде Turbo C 2.0.

Программный продукт содержит: главную управляющую программу в файле ASPPI.EXE;

файл помощи (HELP.IBS);
файл, осуществляющий запуск системы (MAIN.EXE);

систему графической поддержки (файлы ZAST, EGAVGA.EXE) для EGA, VGA, CGA мониторов;

информационные файлы TPEE1.IBS, KATL1.IBS, TEXT1.IBS, KEYS1.IBS, TABL1.IBS в каталогах HOZRAS, CONVER, METODO, INFORM, RESOUR.

Система «Банк Профессиональных Знаний» представляет собой характерный пример современного прикладного программного обеспечения, ориентированного на широкий круг пользователей, не имеющих какой-либо (даже минимальной) подготовки в области использования компьютера. Вся информация, необходимая для работы с системой, представляется на экране в виде

сообщений и подсказок. При необходимости можно воспользоваться помощью — дополнительной информацией, вызываемой системой из файла HELP.IBS.

Система может работать в трех режимах:

1. Режим выбора модуля.
2. Режим печати.
3. Режим поиска информации по ключам.

Управление системой обеспечивается минимальным набором управляющих клавиш. Разработчики стремились придерживаться общепринятого стандарта распределения функций управляющих клавиш.

Для каждого режима работы существует своя строка-подсказка, которая располагается в двух нижних строках экрана.

Система предоставляет пользователям возможность последовательного просмотра всех модулей. Имеется возможность просмотреть аннотации к модулям следующего уровня, а также осуществлять быстрый поиск модуля по заданным ключам. Система позволяет выводить на печать модули текущего и последующего уровней и аннотацию к ним.

Адрес для справок: 103031 Москва, ул. Рождественка, 6/20.

Телефон: 928-87-14.

В. ЛУЦКИЙ, В. ШАХНОВИЧ

Графика «Агата» — новые возможности

В переходный период внедрения IBM-совместимой техники в школу одним из направлений деятельности коллектива, в котором работают авторы, является программно-аппаратное «сближение» ПЭВМ «Агат» с ПЭВМ типа IBM. Наличие схожего по идеологии программного обеспечения и связи ПЭВМ IBM с «Агатом» (текстовые и графические данные) через контроллер RS-232 позволяет легко перенести наработанное программное обеспечение с одной машины на другую и обратно и использовать на учительском месте IBM, а на ученических — «Агаты», объединенные в локальную сеть.

«Диалект» — новые подходы

Возьмем на себя смелость предложить на суд пользователей ПЭВМ «Агат 7/9» новую концепцию текстового редактора «Диалект», рассчитанную на учителя. Пусть каждый выбирает то, что ему нужно:

1. Текст представляется на экране в естественной форме, т. е. так, как он будет распечатан.

2. Пользователь имеет возможность создавать свои шрифты, при этом не надо выходить из текста — шрифт можно редактировать по ходу работы.

3. Текст отображается в графическом режиме, при этом быстрая печать текста на принтере.

4. Удобный оконный интерфейс со сканирующей подсказкой для начинающих и «горячие» клавиши для профессионалов.

5. «Рисование» рамок и таблиц.

6. Совместимость со всеми существующими на «Агате» редакторами.

Естественное представление текста на экране в том виде, в котором он будет распечатан, неудобством не назовешь. При этом текст распечатывается вертикальными полосами, длина строки до 255 символов, независимо от типа принтера (узкий или

широкий). К дополнительным возможностям относится печать одной строки или текста от курсора вниз. Реализованы привычные для работающих на «Агат-авторе» режимы форматирования текста, часть из которых выполняется автоматически в процессе набора текста. Новыми для пользователей будут функции автопереноса, копирование или перенос прямоугольных частей текста, наличие на экране «Агат-7» прописных букв и возможность рисовать таблицы. Таблицы рисуются с помощью клавиш «стрелки».

Предоставляемая пользователю возможность нарисовать свой алфавит реализуется очень просто — в любой момент можно войти в окно редактирования символов, с помощью клавиш «пробел» и «стрелки» изменить символ и продолжить работу с текстом. Это несомненное достоинство текстового редактора «Диалект» позволяет работать как с национальными шрифтами, так и со специальными текстами, насыщенными символами — математическими, физическими и т. п. «Диалект» совместим по идеологии и интерфейсу с текстовыми редакторами типа «Лексикон» и «МИМ» для IBM.

«МаркиС» — рисуют все

Комплекс графических редакторов «МаркиС» практически полностью повторяет идеи известного на IBM графического редактора «Доктор Хало» или аналогичного ему «Доктор Джениус». Редакторы реализованы на «Агат-9» в трех графических режимах (цветном 256×256, черно-белом 512×256 и 256×256) и на «Агате-7» в высокой графике. Интересной проблемой, решенной разработчиками, было подключение серийно выпускаемых манипуляторов «мышь» и «джойстик». Проблема состояла в том, что «Агаты», выпущенные на разных заводах, имеют различную распайку входа «Пульт», к которому подключаются названные устройства. Способ подключения через параллельный интерфейс после обсуждения набрал слишком много «минусов» и принят не был, главным образом из-за удорожания манипуляторов, отсутствия параллельного порта

на «Агат-9» и лишением возможности подключения принтера на «Агат-7».

Итак, в рабочем режиме редактора все функции отображены на боковой и нижней панелях с помощью пиктограмм. Для смены функции необходимо установить курсор на нужную пиктограмму и нажать на кнопку манипулятора «мышь» или клавиатуры.

Такой интерфейс позволяет использовать графический редактор как младшими школьниками, так и опытными программистами для оформления своих программ, а наличие манипулятора «мышь» не только облегчает работу, но и резко повышает качество рисунков.

«Инструмент»

В «ИНФО» не раз обсуждались технологии написания обучающих комплексов с использованием различных библиотек программ и программных оболочек. В предлагаемой работе мы решили совместить первые две технологии, используя разработки «Диалекта» и «МаркиСа». В результате получилась система графических и текстовых редакторов с библиотекой программ и оболочек, позволяющая быстро создавать демонстрационно-обучающие пакеты на языке «Бейсик» как пользователям, не знающим языков программирования, так и программистам с небольшим опытом. Причем если последних не устраивает набор предложенных средств (нельзя же объять необъятное!), то они могут, отдав «Инструменту» рутинную часть работы, легко внести свои изменения и дополнения в окончательные программы. «Инструмент» создает в итоге Бейсик-программу, использующую двоичный файл — библиотеку элементарных графических функций. А наличие в «Инструменте» подпрограмм «Точка», «Линия», «Прямоугольник», «Эллипс» и «Заливка» помогает с комфортом использовать новые графические режимы «Агат-9», которые забыли реализовать в интерпретаторе «Бейсик».

Пользователей ПЭВМ «Агат» приглашаем к сотрудничеству! Телефоны для справок в Москве: 489-00-91, 272-26-71.

Контроль знаний в дисплейном классе

Компьютеры открывают множество интересных возможностей на всех этапах учебного процесса: обучающие и контролирующие программы, подача материала порциями с последующим закреплением, элементы игры и учет индивидуальных возможностей в скорости восприятия материала, тренажеры...

Помимо уроков информатики я (может быть, даже в большей степени) использую компьютеры на других уроках. Основная направленность — контроль знаний учащихся. Хочется сразу внести определенность: для меня компьютерный опрос — дополнения, а не альтернатива известным приемам контроля знаний.

Принцип контроля не нов. По предмету «Физика» имеется большое количество дидактического материала, который и положен в основу контролирующих программ (Д. И. Пеннер и Э. Д. Корж «Программированные задания по физике...» (для IX и X классов), А. В. Постников «Проверка знаний учащихся по физике VI—VII», Л. М. Кузьмин «Сборник вопросов по физике» и многие другие). Построение заданий в этих пособиях следующее: ученик должен найти правильные ответы на предложенные вопросы (4—5 вопросов и столько же ответов). Разумеется, недостатки такого метода очевидны, но как одна из форм контроля годится и он.

Программа для компьютера включает от 10 до 20 вопросов. Каждый вопрос сопровождается правильным ответом и несколькими неверными, но правдоподобными. В начале работы с программой список вопросов выдается на экран. Указывается также норма оценки: какой балл и за какое число верных ответов может получить учащийся. Программа допускает повторные ответы на вопросы, но с условием снижения оценки. Об этом также сообщается в начале работы. Если вопросов больше 10, то учащийся может сам выбрать, на сколько вопросов он желает ответить.

Итак, выбрано количество вопросов; по запросу компьютера в его память введены фамилия, имя учащегося и номер группы. Началась работа программы.

На экране появляются текст вопроса, выбранного случайным образом из предложенного ранее списка, и варианты ответа. Каждый вариант имеет свой номер. Ученику предлагается выбрать нужный вариант, набрав его номер и ввести в память компьютера нажатием на клавишу ВВОД. Компьютер

сравнивает введенное число с числом, соответствующим коду правильного варианта. В случае совпадения на экране высвечивается надпись «ПРАВИЛЬНО» и задается следующий вопрос. Верные ответы накапливаются.

Если введен неверный вариант, то на экране появляется надпись «НЕВЕРНО. ХОЧЕШЬ ПОВТОРИТЬ (Да/Нет)?». При нажатии на клавишу с буквой «Н» компьютер задает следующий вопрос. Количество предложенных вопросов запоминается. Если учащийся хочет повторить ответ на данный вопрос, то на предложение компьютера он должен нажать клавишу с буквой «Д». На экране вновь появляется приглашение набрать и ввести номер выбранного варианта. Число повторных ответов накапливается.

После того как учащийся ответил на все вопросы, компьютер вычисляет оценку, подает звуковые сигналы, число которых соответствует полученному баллу, и выдает на экране информацию о числе заданных вопросов, верных ответов, повторных ответов, снижении оценки за повторные ответы и окончательную оценку за выполненную работу.

Звуковой сигнал по окончании работы дает возможность преподавателю на расстоянии узнавать оценку ученика. Окончательная оценка и фамилия ученика могут быть выведены в информационную строку, что исключает стирание их с экрана при нажатии на клавишу очистки экрана СБР.

```

10 REM===== В.А.ШВАЛЕВ =====
20 ? CHR$(148);CHR$(158);
   CHR$(145);CHR$(12)
30 T=0 *ЧИСЛО ВОПРОСОВ
40 W=0 *ЧИСЛО ВЕРНЫХ ОТВЕТОВ
50 ? AT(0,5) "XXXXXXXXXXXXXXXXXXXXX"
60 ? AT(0,9) " ЭЛЕКТРИЧЕСКИЙ ТОК "
70 ? AT(0,13) "XXXXXXXXXXXXXXXXXXXXX"
80 FOR I=6 TO 12
90 ?AT(0,1)"X";AT(30,1)"X"
100 NEXT I
110 CLEAR 1500
120 DIM X$(15),A$(28),C(28),D(15),
   FR,IMR,GR
130 X$(1)=" ЧТО НАЗЫВАЕТСЯ
   ЭЛЕКТРИЧЕСКИМ ТОКОМ?"
140 A$(1)="...НАПРАВЛЕННОЕ
   ПЕРЕМЕЩЕНИЕ ЭЛЕКТРОНОВ "
150 A$(2)="...НАПРАВЛЕННОЕ ДВИЖЕНИЕ
   ЭЛЕКТРИЧЕСКИХ ЗАРЯДОВ "
160 X$(2)=" ЧТО ПРИНИМАЕТСЯ ЗА
   ТЕХНИЧЕСКОЕ НАПРАВЛЕНИЕ ТОКА?"
170 A$(3)="...НАПРАВЛЕННЫЕ ДВИЖЕНИЯ
   ОТРИЦАТЕЛЬНЫХ ЗАРЯДОВ "
180 A$(4)=MID$(A$(3),1,24)+
   "ПОЛОЖИТЕЛЬНЫХ ЗАРЯДОВ "
```



```

190 X(3)=" ЕДИНИЦА ИЗМЕРЕНИЯ
      ЭЛЕКТРИЧЕСКОГО ЗАРЯДА
200 A(5)="...ОМ"
210 A(6)="...КУЛОН"
220 X(4)=MID(X(3),1,19)+"НАПРЯЖЕНИЯ"
230 A(7)="...АМПЕР "
240 A(8)="...ВОЛЬТ "
250 X(5)=" ПРИБОР ДЛЯ ИЗМЕРЕНИЯ
      НАПРЯЖЕНИЯ"
260 A(9)="...ОММЕТР "
270 A(10)="...ВОЛЬТМЕТР"
280 X(6)=MID(X(5),1,22)+"СИЛЫ ТОКА"
290 A(11)="...ВАТТМЕТР "
300 A(12)="...АМПЕРМЕТР "
310 X(7)=" ФОРМУЛА НАПРЯЖЕНИЯ"
320 A(13)="...U=I/R "
330 A(14)="...U=IXR "
340 X(8)=" ФОРМУЛА СОПРОТИВЛЕНИЯ
      ПРОВОДНИКА"
350 A(15)="...R=U/I "
360 A(16)="...R=U/I "
370 X(9)=" ФОРМУЛА СИЛЫ ТОКА ДЛЯ
      УЧАСТКА ЦЕПИ"
380 A(17)="...I=U/R "
390 A(18)="...I=U/R "
400 X(10)=" ФОРМУЛА ЗАКОНА ОМА ДЛЯ
      ПОЛНОЙ ЦЕПИ"
410 A(19)="...I=U/(R+R') "
420 A(20)="...I=E/(R+R') "
430 X(11)=" СОПРОТИВЛЕНИЕ ПРОВОДНИКА
      ПРИ НАГРЕВАНИИ..."
440 A(21)="...НЕ ИЗМЕНЯЕТСЯ "
450 A(22)="...УВЕЛИЧИВАЕТСЯ "
460 X(12)=" ПРОВОДИМОСТЬ МЕТАЛЛОВ "+
      MID(X(11),27)
470 A(24)="...УМЕНЬШАЕТСЯ "
480 X(13)=" ДЛЯ ЧЕГО СЛУЖИТ РЕОСТАТ?"
490 A(25)="...ДЛЯ ИЗМЕРЕНИЯ
      СОПРОТИВЛЕНИЯ УЧАСТКА ЦЕПИ"
500 A(26)="ДЛЯ ИЗМЕНЕНИЯ
      СОПРОТИВЛЕНИЯ ЭЛЕКТРИЧЕСКОЙ ЦЕПИ"
510 X(14)=" ДЛЯ ЧЕГО СЛУЖИТ АМПЕРМЕТР?"
520 A(27)="...ДЛЯ ИЗМЕРЕНИЯ НАПРЯЖЕНИЯ"
530 A(28)="...ДЛЯ ИЗМЕРЕНИЯ СИЛЫ ТОКА"
540 X(15)=X(5)
550 ? AT(8,21)"НАЖМИТЕ 'ВВОД' "
560 IF INKEY<>CHR(10) THEN 560
570 ? AT(0,19);CHR(153)"ВАША ФАМИЛИЯ";
580 INPUT F(1)
590 IF F(1)="" THEN 570
600 ? AT(0,19);CHR(153)"ИМЯ";
610 INPUT I(1)
620 IF I(1)="" THEN 600
630 ? AT(0,19);CHR(153)"НОМЕР ГРУППЫ";
640 INPUT G(1)
650 IF G(1)="" THEN 630
660 CLS
670 ? CHR(155)"ВАМ НУЖНО ОТВЕТИТЬ
      НА СЛЕДУЮЩИЕ ВОПРОСЫ:"
680 ?
690 FOR I=1 TO 15
700 ? I;X(I)
710 NEXT I
720 ? CHR(156)"ЕСЛИ ГОТОВЫ ОТВЕЧАТЬ,";
730 ? "НАЖМИТЕ 'ПРОБЕЛ' ";CHR(156)
740 IF INKEY<>" " THEN 740
750 CLS
760 ? AT(0,3);CHR(155)"НА СКОЛЬКО
      ВОПРОСОВ БУДЕТЕ"
770 ? "ОТВЕЧАТЬ, ";I;X;"?"
780 ? AT(0,6)"НАБЕРИТЕ ЧИСЛО
      И НАЖМИТЕ 'ВВОД'"

```

```

790 ? AT(0,8);CHR(153)"ВАШЕ ЧИСЛО
      (ОТ 5 ДО 15)";
800 INPUT B(1)
810 IF B(1)="" OR VAL(B(1))=0 THEN 790
820 B=VAL(B(1))
830 IF B>15 OR B<5 THEN 790
840 ? CHR(155)
850 FOR I=B TO 1 STEP -1
860 OC=INT(((5*I)/B)+0.7)
870 IF OC=1 THEN 890
880 ? "ОЦЕНКА ";OC;" ЗА ";I;" ВЕРНЫХ
      ОТВЕТА"
890 NEXT I
900 ? "ЗА ПОВТОРНЫЕ ОТВЕТЫ
      ОЦЕНКА СНИЖАЕТСЯ"
910 ? CHR(156)"ЕСЛИ ГОТОВЫ ОТВЕЧАТЬ,";
920 ? " НАЖМИТЕ 'ПРОБЕЛ' ";CHR(156)
930 Y=INT(RND(1)*15+1)
940 IF INKEY<>" " THEN 940
950 T=T+1
960 FOR U=1 TO 28
970 C(U)=0
980 NEXT U
990 CLS
1000 ? CHR(156)" ВОПРОС";T;
      CHR(156);X(Y)
1010 IF Y=15 THEN 1790
1020 ?
1030 M=2*X-5
1040 N=2*X+7
1050 IF M<=0 THEN M=1
1060 IF N>28 THEN N=28
1070 K=INT(RND(1)*N+1)
1080 ? "..... О Т В Е Т Ы ....."
1090 FOR J=M TO N
1100 F=INT(RND(1)*N+1)
1110 IF F<M THEN 1100
1120 IF C(F)=1 THEN 1100
1130 C(F)=1
1140 IF LEN(A(F))=0 THEN 1170
1150 KOD=LEN(A(F))+K
1160 ? CHR(155)KOD;CHR(155)A(F)
1170 NEXT J
1180 OTW=LEN(A(2*X))+K
1190 ? CHR(156)
1200 ? "НАБЕРИТЕ НОМЕР ОТВЕТА,";I;X;
      ", И НАЖМИТЕ 'ВВОД' ";
1210 ? CHR(156);CHR(155)
1220 S=CSRLIN(0)
1230 GOTO 1310
1240 P=P+1
1250 ? AT(0,22)" БЫЛ";
1260 FOR K=21 TO S STEP -1
1270 LOCATE 0,K
1280 ? CHR(19);CHR(7);
1290 NEXT K
1300 S=S+1
1310 ? AT(0,22);CHR(153);"ВАШ ОТВЕТ:";
1320 N(1)=""
1330 INPUT N(1)
1340 IF N(1)="" OR VAL(N(1))=0 THEN 1310
1350 IF VAL(N(1))=OTW THEN 1460
1360 ? AT(0,21)"НЕВЕРНО, БУДЕШЬ
      ПОВТОРЯТЬ?";
1370 ? CHR(156)" Д/Н ";CHR(156)
1380 A(1)=INKEY(1)
1390 IF A(1)=CHR(10) THEN 1380
1400 IF A(1)="" THEN 1380
1410 IF A(1)="D" OR A(1)="d" OR
      A(1)="Д" OR A(1)="д" THEN 1240
1420 IF A(1)="N" OR A(1)="n" OR A(1)="Н"
      OR A(1)="н" THEN 1440 ELSE 1380

```

```

1440 ? CHR$(155)
1450 GOTO 1480
1460 ? "АТ(0,21)" П Р А В И Л Ь Н О !";
CHR$(155)

```

```

1470 W=W+1
1480 FOR JJ=1 TO 500
1490 NEXT JJ
1500 D(Y)=1
1510 IF T=W THEN 1550
1520 Y=INT(RND(1)*15+1)
1530 IF D(Y)=1 THEN 1520
1540 GOTO 950
1550 OC=INT(((5*W)/B)+0.7)
1560 IF B>5 AND B<=10 THEN OS=OC-(P\3)
1570 IF B>10 THEN OS=OC-(P\4) ELSE
OS=OC-(P\2)

```

```

1580 IF OS<2 THEN OS=2
1590 FOR I=1 TO OS
1600 FOR A=1 TO 200
1610 POKE -50,-1
1620 POKE -50,1
1630 NEXT A
1640 FOR J=1 TO 20
1650 NEXT J,I
1660 CLS

```

52

```

1670 ? CHR$(155)АТ(0,5)"ЗАДАНО
ВПРОСОВ ";B

```

```

1680 ?
1690 ? "ВЕРНЫХ ОТВЕТОВ:"W
1700 ? "НЕВЕРНЫХ:";B-W
1710 ? "П О В Т О Р Н Ы Х:";P
1720 ? "СНИЖЕНИЕ ЗА ПОВТОР

```

```

(В БАЛЛАХ):";OC-OS
1730 ? АТ(0,15)F$;" ";I$;" ГР.";G$
1740 ? CHR$(156)" О Ц Е Н К А : ";
OS;" ";CHR$(156)

```

```

1750 POKE 112,15360+PEEK(132)
1760 ? CHR$(148);CHR$(158);CHR$(145);
1770 ? CHR$(156);OS;F$;CHR$(156)
1780 END
1790 ? CHR$(155); АТ(4,7)"А

```

```

1800 FOR I=35 TO 227 STEP 64
1810 CIRCLE (I,75),15,,,1.4
1820 LINE (I-20,75)-(I-10,75)
1830 LINE (I+10,75)-(I+20,75)
1840 NEXT I
1850 OTW=2
1860 ? АТ(4,10)"1 2 3
4"; CHR$(155)
1870 GOTO 1190

```

Рассмотрим подробнее работу программы.

Строка 20 — очистка служебной строки. Строки 110—120 — резервирование памяти под символьные и числовые переменные. X (15) — массив вопросов, A (28) — массив вариантов ответов (на каждый вопрос может быть два варианта ответа; второй — верный). C(28) — массив, номера элементов которого соответствуют порядковым номерам вариантов ответов. Элементами этого массива являются числа 0 (если данный ответ не выведен на экран) и 1 (если ответ выводится на экран) (строки 1100, 1120, 1130). Перед выводом на экран очередного вопроса элементы массива C(28) обнуляются (см. строки 960—980). Работая с элементами этого массива, компьютер не допускает повто-

ряемости одного и того же ответа при выводе вариантов ответов на заданный вопрос (см. строки 1100, 1120, 1160).

Номера элементов массива D(15) соответствуют номерам вопросов. Элементами массива являются числа 0 (если соответствующий вопрос еще не задан) и 1 (если вопрос уже задан) (строки 930 и 1520 — выбор номера вопроса, 1530 — проверка на неповторяемость).

Строки 130—540 — список вопросов и ответов. Некоторые неверные ответы можно опускать (например, на вопрос X (12) нет одного ответа).

Формула оценки (строка 860) может быть иной.

Строка 930 — выбор номера первого вопроса, 1000 — вывод вопроса на экран, 1030, 1040 — задание верхней и нижней границ номеров вариантов ответов, которые будут выведены на экран.

Строки 1080—1170 — вывод на экран списка ответов, 1120, 1130 — контроль на неповторяемость, 1140 — проверка на несуществующий ответ. Нумерация ответов на экране (1150) производится следующим образом: подсчитывается длина символьной строки ответа, к полученному числу прибавляется заранее заданное случайное число (строка 1070). Строка 1180 — номер верного ответа. Чтобы исключить равенство длин символьных строк, в ответы добавляются лишние пробелы.

1320—1350 — запрос номера ответа и анализ введенного числа. Запрашивается символьная величина (!), как и в строке 800, для страховки возможной ошибки учащегося при вводе числа с клавиатуры. В случае верного ответа подсчет числа верных ответов (1470), задержка (1480, 1490), запоминание номера заданного вопроса (1500), проверка исчерпания числа вопросов (1510), выбор нового вопроса (1520), проверка на неповторяемость (1530).

В случае неверного ответа переход к строке 1240, суммирование повторного ответа.

Строки 1550—1780 — обработка результатов опроса, окончание работы. 1550—1580 — формирование оценки. 1590—1650 — звуковой сигнал об окончании работы. 1760, 1770 — запись в служебную строку.

Некоторые вопросы (как, например, X (15)) могут иметь ответы в особой подпрограмме (строки 1790—1870) с чертежами, рисунками.

Программы, подобные приведенной, используются преподавателями на уроках физики, астрономии, сельхозмашин, истории и других предметов.

Программа легко может быть переписана для любого компьютера. Недостатком ее яв-

ляется необходимость работы в режиме 64 символа в строке (ввиду большого объема текстовой информации), что невозможно при работе с цветным монитором.

К сожалению, отношение профессиональных программистов к подобным программам более чем равнодушное. Чаще всего такие программы пишутся самими преподавателями и учениками, причем каждый начинает с изобретения велосипеда, так как «ИНФО» ни разу не предложил программ подобного уровня. Имеющиеся учебные программы на Т-языке не лучше уже по той причине, что преподавателю необходимо владеть этим языком, чтобы адаптировать имеющуюся программу или написать новую.

Разработчики рекламируют программы в кодах, забывая при этом пользователя. Компьютеры приобретаются уже и сельскими

школами, но кто работает на них? Преподаватели со стажем, ведущие физику, математику и имеющие зачастую только представление о Бейсике. Педвузы только начинают готовить преподавателей информатики. Компьютеры стоят без программного обеспечения. В лучшем случае (а я считаю — в худшем) используются как игровые автоматы.

Предлагаемая программа при всей ее простоте хороша уже тем, что является универсальной. Замените вопросы по физике вопросами по истории, литературе, дайте эту программу учителю географии — пусть впишет свои вопросы, и каждый сделает десять своих программ для себя, для своих учеников, для своих Сидорова, Петрова, Иванова. Пусть это будет тренажер — он ведь тоже необходим. Учителю нужны и такие программы.

С. СОЧНЕВ

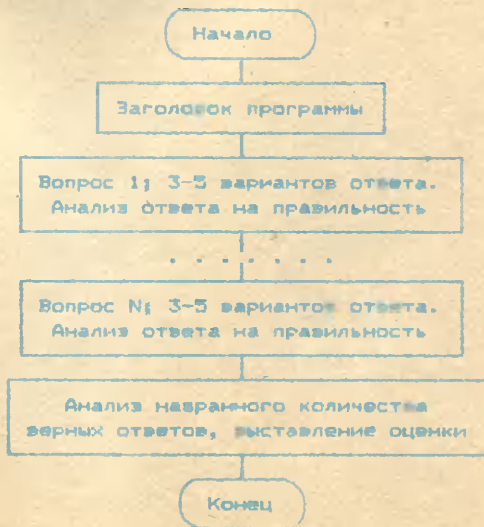
Пример контролирующей программы

В статье рассматриваются принципы построения и ряд особенностей реализации контролирующих программ в дисплейном классе УКНЦ (МС0202). Этот материал может быть использован при самостоятельном создании программного обеспечения для изучения предметов школьного курса (не только информатики). После внесения изменений, связанных с различиями версий языка Бейсик, приводимые в работе фрагменты программ могут быть использованы и в других дисплейных классах.

Принципы построения контролирующих программ

Контроль знаний предполагает ответы ученика на ряд вопросов с выставлением оценки. Этой цели может служить программа, структура которой представлена на блок-схеме.

Заголовок программы обычно содержит названия изучаемого курса, темы и занятия, краткую инструкцию по работе, рассчитанную на неподготовленного пользователя. Кроме того, здесь надо предусмотреть ввод информации об ученике, работающем за дисплеем (как минимум его фамилию). Контролирующая программа содержит обычно 5—10 вопросов. В дальнейшем мы будем рассматривать программы с пятью вопросами, тогда количество верных ответов станет оценкой ученику за выполненную работу. Анализ ответа на вопрос



(правильен или нет) удобнее выполнять с помощью подпрограммы, так как эта операция повторяется после каждого вопроса. Ответы на вопросы и их анализ можно организовать следующим образом.

Фрагмент 1

```

10 K=0
15 REM K - КОЛИЧЕСТВО ВЕРНЫХ ОТВЕТОВ
. . .
100 REM ВОПРОС НОМЕР N
110 CLS
120 ? AT(5,2)"ТЕКСТ ВОПРОСА
130 ? AT(5,4)"1 - ПЕРВЫЙ ВАРИАНТ ОТВЕТА
140 ? AT(5,5)"2 - ВТОРОЙ ВАРИАНТ ОТВЕТА
. . .
180 F=F
185 REM P - КОНКРЕТНОЕ ЧИСЛО, НОМЕР
      ВЕРНОГО ОТВЕТА
190 GOSUB 1000
200 REM СЛЕДУЮЩИЙ ВОПРОС
. . .
1000 REM ПОДПРОГРАММА АНАЛИЗА ОТВЕТА
1010 ? AT(10,20) "КАКОВ НОМЕР
      ПРАВИЛЬНОГО ОТВЕТА";
54 1020 INPUT M
1030 CLS
1040 IF M=F THEN 1070
1050 ? AT(26,10) "ВЫ ОШИБЛИСЬ
1060 GOTO 1090
1070 ? AT(27,10) "ПРАВИЛЬНО
1080 K=K+1
1090 FOR I=1 TO 5000
1100 NEXT I
1110 RETURN

```

Для вывода информации используется функция AT, в других версиях Бейсика можно применить оператор LOCATE. Строки 1090—1100 формируют задержку в 3—4 с, чтобы слова «ВЫ ОШИБЛИСЬ» или «ПРАВИЛЬНО» не пропадали с экрана мгновенно.

Более рационально для ввода номера правильного ответа использовать оператор INKEY\$, так как в этом случае не надо подтверждать каждый ответ нажатием клавиши «ВВОД». Тогда вместо строки 1020 следует записать

Фрагмент 2

```

1020 X$=INKEY$
1021 IF X$="" THEN 1020
1022 M=VAL(X$)

```

Отметим, что у неподготовленного пользователя нередки случаи, когда, нажав клавишу, он держит ее несколько секунд. Вместо номера правильного ответа «2» набирается «22222...». Первую цифру «2» ЭВМ воспринимает как ответ на данный вопрос, следующую «2» — как ответ на следующий вопрос и т. д. Перед учеником один за другим пробегают вопросы программы... Чтобы избежать этого, подпрограмму следует удлинить.

Фрагмент 3

```

1023 FOR I=1 TO 5000
1024 NEXT I

```

```

1025 X$=INKEY$
1026 IF X$<>"" THEN 1025

```

Первые две строки дают задержку в 3—4 с. Предполагается, что за это время ученик успеет отпустить нажатую клавишу. Две следующие строки «гасят» лишние цифры и позволяют не учитывать их при переходе к очередному вопросу. В итоге после перенумерации строк подпрограмма примет вид

Фрагмент 4

```

1000 REM ПОДПРОГРАММА АНАЛИЗА ОТВЕТА
1010 ? AT(10,20) "КАКОВ НОМЕР
      ПРАВИЛЬНОГО ОТВЕТА";
1020 X$=INKEY$
1030 IF X$="" THEN 1020
1040 M=VAL(X$)
1050 FOR I=1 TO 5000
1060 NEXT I
1070 X$=INKEY$
1080 IF X$<>"" THEN 1070
1090 CLS
1100 IF M=F THEN 1130
1110 ? AT(26,10) "ВЫ ОШИБЛИСЬ
1120 GOTO 1150
1130 ? AT(27,10) "ПРАВИЛЬНО
1140 K=K+1
1150 FOR I=1 TO 5000
1160 NEXT I
1170 RETURN

```

За каждым РМУ — свой вариант программы

Недостатком рассмотренной программы является возможность подскоков, так как за всеми РМУ вопросы и номера правильных ответов совпадают. Ликвидировать его можно перестановкой вариантов ответов так, чтобы правильный ответ на разных РМУ имел разные номера. Наиболее рациональный путь для этого — присвоить варианты ответа элементам символьного массива (например, первый вариант занесем в A\$(1), второй в A\$(2) и т. д.) и организовать какую-либо перестановку этих элементов, например произвести сдвиг влево или вправо на K номеров, где K будет определяться с помощью генератора случайных чисел RND. При сдвиге вправо на один номер элементы массива из пяти ответов расположатся в следующем порядке:

A\$(5) A\$(1) A\$(2) A\$(3) A\$(4)

а при сдвиге влево на один номер —

A\$(2) A\$(3) A\$(4) A\$(5) A\$(1)

Однако оказалось, что использование пяти символьных переменных достаточной длины (1—2 экранные строки) для класса УКНЦ практически невозможно из-за малого объема памяти, выделяемой для символьных переменных. В то же время для текста

программы выделяется значительно больший объем памяти, что позволяет организовать перестановку вариантов ответа без использования символьных величин.

Фрагмент 5

```
100 REM ВОПРОС НОМЕР N
110 CLS
120 ? AT(5,2) "ТЕКСТ ВОПРОСА
130 W=INT(RND(1)*4.99+1)
140 V=1
150 ON W GOTO 160,200,230,260,290,320
160 ? AT(0,6+2*V) V " - ПЕРВЫЙ
                                ВАРИАНТ ОТВЕТА
170 F=V
180 GOSUB 2000
190 GOTO 150
200 ? AT(0,6+2*V) V " - ВТОРОЙ
                                ВАРИАНТ ОТВЕТА
210 GOSUB 2000
220 GOTO 150
230 ? AT(0,6+2*V) V " - ТРЕТИЙ
                                ВАРИАНТ ОТВЕТА
240 GOSUB 2000
250 GOTO 150
260 ? AT(0,6+2*V) V " - ЧЕТВЕРТЫЙ
                                ВАРИАНТ ОТВЕТА
270 GOSUB 2000
280 GOTO 150
290 ? AT(0,6+2*V) V " - ПЯТЫЙ
                                ВАРИАНТ ОТВЕТА
300 GOSUB 2000
310 GOTO 150
320 GOSUB 1000
200 REM СЛЕДУЮЩИЙ ВОПРОС
.
.
.
1000 REM ПОДПРОГРАММА АНАЛИЗА
                                ОТВЕТА (СМ. ФРАГМЕНТ 4)
.
.
.
2000 REM ПОДПРОГРАММА
2010 V=V+1
2020 W=W+1
2030 IF W=6 THEN W=1
2040 IF V=6 THEN W=6
2050 RETURN
```

Переменная W, определяемая в строке 130, является номером варианта ответа, который станет первым после перестановки (возможные значения — от 1 до 5). Переменная V определяет номера вариантов ответа после перестановки («новые» номера). Строка «F=V» ставится после правильного ответа, в данном фрагменте таким является первый (см. строку 170). Имеет смысл всегда ставить правильный ответ первым, так как программа сама осуществит его перестановку в любое место последовательности ответов.

Заметим, что при запуске такой программы на нескольких РМУ непосредственно после их включения перестановки вариантов ответов будут везде одинаковы (т. е. номер правильного ответа хотя и изменяется, но для разных РМУ он один и тот же). Действительно, генератор RND на всех ЭВМ запустился одинаково и W получают везде одни и те же значения.

Очевидно, при начале работы программы необходимо произвести запуск генератора RND в зависимости от некоторой случайной величины. В качестве такой величины удобно взять вторую букву фамилии ученика:

Фрагмент 6

```
30 INPUT "ФАМИЛИЯ";FX
40 ZX=MID$(FX,2,2)
50 Z=ASC(ZX)
60 FOR I=1 TO Z
70 R=RND(1)
80 NEXT I
```

Отметим, что без набора фамилии программа в этом случае работать не будет, так как выполнение строки 40 приведет к ошибке.

Наряду с указанным способом создания различных вариантов ответов возможно программирование нескольких параллельных вопросов (вопросы 1—А, 1—Б, 1—В), из которых с помощью генератора случайных чисел выбирается один. В сочетании с перестановкой ответов это дает свой вариант программы для каждого РМУ. Память ЭВМ позволяет разместить 5—10 вопросов в двух вариантах (вопросы 1—А и 1—Б, 2—А и 2—Б и т. д.) в зависимости от длины самих вопросов, вариантов ответов и заголовка программы.

Ограничение времени

Нередко возникает необходимость ограничить время работы ученика с контролирующей программой 10—15 мин, т. е. для ответа на вопрос выделить не более 1—2 мин. Ограничение времени можно осуществить программно, используя в качестве величины, следящей за затратами времени, количество проходов по кругу

```
1020 XX=INKEY$
1030 IF XX="" THEN 1020
```

до нажатия учеником какой-либо клавиши. В этом случае подпрограмма анализа ответа (фрагмент 4) несколько усложнится.

Фрагмент 7

```
1000 REM ПОДПРОГРАММА АНАЛИЗА ОТВЕТА
1005 S=0
1010 ? AT(20,10) "КАКОВ НОМЕР
                                ПРАВИЛЬНОГО ОТВЕТА";
1020 XX=INKEY$
1021 S=S+1
1022 IF S<RRR THEN 1030
1023 CLS
1024 ? AT(24,10) "ВРЕМЯ ЗАКОНЧИЛОСЬ
1025 GOTO 1150
1030 IF XX="" THEN 1020
1040 REM СТРОКИ 1040-1170
                                БЕЗ ИЗМЕНЕНИЙ (СМ. ФРАГМЕНТ 4)
```

Переменная S играет роль счетчика времени. В зависимости от того, какое время выделяется для ответа на вопрос, подбирается значение величины RRR (чем оно больше, тем больше время). Значение RRR подбирают методом проб, оно обычно лежит в пределах нескольких тысяч — десятков тысяч. При этом следует учитывать, что быстроедействие программы на РМУ и РМП в DBAS различно, поэтому подбор надо выполнять на РМУ. Кроме того, объем памяти для РМП в DBAS значительно меньше, чем в РМУ, поэтому составление и отладку программы также лучше выполнять на РМУ.

Задачи

Использование задач в контролирующих программах более целесообразно, так как при выборе варианта правильного ответа присутствует элемент угадывания. При работе с задачами нет необходимости выводить несколько вариантов ответа, численный ответ после вычислений будет введен учащимися по оператору INPUT. Задачи целесообразно варьировать с помощью RND. Приведем пример.

Фрагмент В

```
100 REM ЗАДАЧА "ПЛОЩАДЬ ПРЯМОУГОЛЬНИКА"
110 A=INT(RND(1)*9.9+1)
120 B=INT(RND(1)*4.9+1)
130 ? AT(5,2) "НАЙДИТЕ ПЛОЩАДЬ
    ПРЯМОУГОЛЬНИКА СО СТОРОНАМИ "
    A " И " B
140 F=AXB
150 GOSUB 1000
160 REM СЛЕДУЮЩАЯ ЗАДАЧА
```

```
1000 REM ПОДПРОГРАММА АНАЛИЗА
    ОТВЕТА (СМ. ФРАГМЕНТ 1)
```

Здесь длина стороны А может принимать значения от 1 до 10, длина стороны В — от 1 до 5; имеются 50 вариантов численных данных. Анализ ответа выполняется с помощью подпрограммы, использующей оператор INPUT (фрагмент 1); использование оператора INKEY $\$$ нецелесообразно, так как ответ может содержать разное количество цифр.

При подборе вариантов численных данных следует стремиться, чтобы при решении задачи не получалось приближенных результатов, ведь округленный ответ ЭВМ воспримет как неверный.

При решении задач в ряде случаев возникает необходимость выполнения учащимися вычислений. В некоторых ЭВМ оператор INPUT допускает ввод не только чисел, но и арифметических выражений,

однако в классе УКНЦ такая возможность не предусмотрена. Тем не менее необходимые вычисления можно выполнить на ЭВМ, предусмотрев такой вариант работы в самой контролирующей программе. При работе на ЭВМ ученику предоставляется возможность набрать ответ (некоторое число) или число 0, тем самым переводя ЭВМ в режим вычислений (при такой организации программы следует отказаться от задач с нулевым ответом). После выполнения вычислений ЭВМ вновь спрашивает ответ, и, если ученик вновь наберет 0, ЭВМ снова перейдет в режим вычислений. Для реализации такого режима следует изменить подпрограмму анализа ответа.

Фрагмент 9

```
1000 REM ПОДПРОГРАММА АНАЛИЗА ОТВЕТА
1010 ? AT(9,20) "ДЛЯ ВЫПОЛНЕНИЯ
    ВЫЧИСЛЕНИЙ НАЖМИТЕ '0' И 'ВВОД'
1020 ? AT(14,21) "ИНАЧЕ НАБЕРИТЕ ОТВЕТ
    И НАЖМИТЕ 'ВВОД'
1030 ? AT(30,22) " ";
1040 INPUT M
1050 IF M<>0 THEN 1240
1060 REM РЕЖИМ ВЫЧИСЛЕНИЙ
1070 FOR I=20 TO 23
1080 ? AT(0,1) SPC(60)
1090 NEXT I
1100 ? AT(23,20) "РЕЖИМ ВЫЧИСЛЕНИЙ
1110 INPUT"ЧЕМУ РАВНО ПЕРВОЕ ЧИСЛО ";T1
1120 INPUT"КАКОЕ ДЕЙСТВИЕ ВЫПОЛНИТЬ";T2
1130 INPUT"ЧЕМУ РАВНО ВТОРОЕ ЧИСЛО";T3
1140 IF T2="+" THEN T3=T1+T2
1150 IF T2="-" THEN T3=T1-T2
1160 IF T2="*" THEN T3=T1*T2
1170 IF T2="/" THEN T3=T1/T2
1180 IF T2="^" THEN T3=T1^T2
1190 FOR I=18 TO 23
1200 ? AT(0,1) SPC(60)
1210 NEXT I
1220 ? AT(1,18) T1; T2; T3; "="; T3
1230 GOTO 1010
1240 CLS
1250 IF M=F THEN 1280
1260 ? AT(26,10) "ВЫ ОШИБЛИСЬ
1270 GOTO 1300
1280 ? AT(27,10) "ПРАВИЛЬНО
1290 K=K+1
1300 FOR I=1 TO 5000
1310 NEXT I
1320 RETURN
```

Перед началом работы учащихся необходимо познакомиться с соответствующими клавишами. Число выполняемых действий можно легко расширить.

Анализ и вывод результата работы программы

Для выставления оценки необходимо выработать критерий (за какое количество верных ответов ставится та или иная оценка). В случае пяти вопросов оценка может совпадать с количеством верных ответов К. Вывод результата состоит в этом слу-

чае из одного оператора PRINT K. Этот же результат можно вывести и на печатающее устройство у преподавателя:

```
Фрагмент 10
900 OPEN "LP:RES" FOR OUTPUT
910 PRINT# F#; " ОЦЕНКА "; K
920 CLOSE
```

Изменив LP на TT, можно выводить результаты работы на дисплей преподавателя. РМП в обоих случаях должно быть в режиме связи класса УКНЦ («Продолжение»). Возможен вариант, когда без указания фамилии в строке 30 (фрагмент 6) результат к преподавателю поступать не будет.

```
Фрагмент 11
30 INPUT "ФАМИЛИЯ";F#
31 IF F#<>" " THEN 40
```

Н. СОФРОНОВА
СШ № 29, г. Чебоксары

Использование графических возможностей «Синклер Спектрум»

Достаточно богатые графические возможности все шире распространяющегося в школах домашнего компьютера «Спектрум» позволяют уже на первых уроках информатики решать кое-какие задачи по программированию. Достаточно ознакомить учеников с клавиатурой и операторами Бейсика PLOT, DRAW, CIRCLE, чтобы они могли рисовать на экране. После подобного разбора только одной картинке многие приносят на следующий урок самостоятельно составленные программы, например, такие:

```
5 REM "ЗОНТИК"
10 PLOT 70,80: DRAW 100,0
20 PLOT 120,10: DRAW 0,120
30 PLOT 70,80: DRAW 80,0,-PI
40 PLOT 90,80: DRAW 80,0,-PI
50 PLOT 120,10: DRAW 20,0,PI
Или
10 PLOT 125,40: DRAW 0,100
20 PLOT 40,40: DRAW 160,0
30 DRAW -10,-10: DRAW -140,0: DRAW -10,10
40 PLOT 125,135: DRAW -20,-40
                    DRAW 20,-40
50 PLOT 125,135: DRAW -60,-40:
                    DRAW 60,-40
60 CIRCLE 210,150,20
70 PLOT 0,30: DRAW 20,0
80 PLOT 210,30: DRAW 15,0
90 PLOT 10,20: DRAW 30,0
100 PLOT 30,10: DRAW 40,0
```

```
32 Y=1
33 GOTO 90
40 REM СТРОКИ 40-80 ИЗ ФРАГМЕНТА 6
. . .
890 IF Y=1 THEN 930
900 OPEN "LP:RES" FOR OUTPUT
910 PRINT# F#; " ОЦЕНКА "; K
920 CLOSE
930 ...
```

При пустой фамилии строки 40—80 будут пропущены, так как выполнение функции MID\$(F#,2,2) в этом случае приводит к ошибке и прекращению работы программы. Такой вариант работы целесообразно использовать при отработке решения задач. Несколько раз прорешав все задачи и хорошо разобравшись в них, ученик наконец набирает свою фамилию и теперь уже работает на оценку. При работе с теоретическими вопросами, ответы на которые даются в нескольких вариантах, работа в таком режиме теряет смысл.

```
110 PLOT 100,15: DRAW 40,0
120 PLOT 180,9: DRAW 30,0
Позднее, когда учащиеся уже знакомы с циклом, можно рисовать картинки типа «спираль» или «рог изобилия».
10 REM РОГ ИЗОБИЛИЯ
20 FOR I=1 TO 50
30 CIRCLE 130+100*I*COS(I/20),
                    20+80*I*SIN(I/20), I
40 NEXT I
```

Есть и другой способ построения изображений — с помощью оператора PRINT и модификатора AT в графическом режиме работы компьютера. В следующей программе использованы также оператор цвета INK и мерцания FLASH.

```
2 INK 6: FOR I=1 TO 128
3 PLOT I,0: DRAW 0,175
4 NEXT I
5 INK 1: FOR I=129 TO 255
6 PLOT I,0: DRAW 0,175
7 NEXT I
10 INK 2: FLASH 1
20 PRINT AT 16,16; " _____"
30 PRINT AT 15,15; " _____"
40 PRINT AT 14,13; " _____"
50 PRINT AT 13,11; " _____"
60 PRINT AT 12,12; " _____"
70 PRINT AT 11,13; " _____"
80 PRINT AT 10,14; " _____"
90 PRINT AT 9,16; " _____"
```

```

100 FLASH 0: INK 6
110 PRINT AT 12,17;"—"
120 INK 4:PLOT 200,0:DRAW -40,38
130 PRINT AT 2,7;INK 3;

```

"Аленький цветочек"

Раскрашивать картинки на «Спектруме» несколько сложнее, чем на БК-0010. В «Аленьком цветочке» сначала рисуется двухцветный фон. Этот же прием можно использовать при составлении программы «Столбиковые диаграммы». Число столбиков, их высота и название диаграммы задаются пользователем в диалоге.

```

10 INPUT "Сколько данных";N
20 DIM A(N):PRINT "Запишите данные"
30 FOR I=1 TO N
40 INPUT A(I)
50 NEXT I:CLS
60 PRINT "Как будет называться
   диаграмма":INPUT R$:CLS
70 PRINT AT 0,7;R$
80 FOR I=1 TO N
90 FOR K=0 TO 20
100 INK I:PLOT 200*I+K,0:DRAW 0,A(I)
110 NEXT K:NEXT I

```

При изучении темы «Логические элементы ЭВМ» тоже можно использовать графическую интерпретацию. Известна большая серия задач типа «Принадлежит ли точка фигу-

ре?». Обычно в них требуется лишь получить ответ, но небольшое усложнение программы позволяет сделать задачу более зрелищной и информативной.

```

10 INPUT "Запишите координаты";X,Y
20 PLOT 120,0:DRAW 0,175
30 PLOT 0,80:DRAW 255,0
40 CIRCLE 120,80,20
50 CIRCLE 120,80,40
60 IF XX+YY=1 AND XX+YY<=4 THEN
   PRINT "Принадлежит";:GOTO 80
70 PRINT "Не принадлежит"
80 LET X=X*20+120
90 LET Y=Y*20+120
100 PLOT X,Y

```

В заключение — два примера изображения кривых поверхностей.

```

10 FOR K=1 TO 8 STEP .5
20 FOR I=1 TO 2*PI STEP .1
30 LET Y=3*K*X SIN(I)
40 LET X=I*10+10*K
50 LET Y=Y*3+80
60 PLOT X,Y
70 NEXT I:NEXT K

10 REM УСЕЧЕННЫЙ КОНУС
20 FOR K=1 TO 7 STEP .5
30 CIRCLE 120,50+K*10,50-K*5
40 NEXT K

```

М. ФЕДОРОВА

Физиологическое обоснование режима обучения школьников при работе на ЭВМ

Внедрение компьютеризации в процесс обучения требует объективной оценки функционального состояния организма учащихся. Исследованиями последних лет показано, что занятия в дисплейном классе более утомительны, чем обычные уроки, и требуют рациональной организации учебной деятельности [1, 4, 5].

Целью проведенной работы было изучение особенностей гемодинамических реакций у юношей 15—17 лет, занимающихся по программе «лаборант-программист» в УПК Сокольнического р-на Москвы. Было проведено обследование 70 юношей — учащихся X и XI классов, занимающихся в кабинете, оборудованном компьютерами «Ямаха-MSX1». В эксперименте были заняты школьники I и II группы здоровья. Занятия проводились 1 раз в неделю в две смены — с 8 ч 30 мин и 13 ч 30 мин.

40 учащихся обследовали 2 раза в течение года — в I и IV четверти. Показатели снимались трижды: в начале занятия, через 2 ч и 4 ч занятий. В III четверти у 30 испытуемых в условиях естественного эксперимента регистрировались показатели в ходе работы на ЭВМ. Исследовались показатели мозгового кровообращения с помощью фокусирующего и монополярного реоэнцефалографического методов, определялся ударный объем сердца по Кубичеку, измерялось артериальное давление. В ходе эксперимента проводилось тестирование обследуемых на эмоциональную стабильность-нестабильность по методике Айзенка в модификации В. М. Русалова (1987) и на мануальную асимметрию по методике В. Ю. Вильдавского и М. Г. Князевой (1987). Результаты исследования обрабатывались методом вариационной статистики

на ЭВМ АСС-8000.

Сравнение особенностей реакций центрального и мозгового кровообращения на умственную нагрузку при работе в дисплейном классе у учащихся, занимающихся по программе разной сложности, показало, что выраженность изменений зависит от вида занятий. У обучающихся по курсу ОИВТ (1 раз в неделю по 2 ч) регистрировались изменения только со стороны показателей системного артериального давления. Диастолическое давление (ДАД) увеличилось на 6,7 % ($p < 0,05$), среднее давление (ср. АД) — на 4,4 % ($p < 0,05$). У девятиклассников, занимающихся по специальности «лаборант-программист» (2 ч теоретических занятий и 2 ч самостоятельной работы на ЭВМ), наблюдались выраженные изменения показателей системной гемодинамики и повышение тонуса мозговых артериол. Сравнение теоретических занятий в дисплейном классе с непосредственной работой на персональных ЭВМ показало, что практические занятия вызывают более выраженное повышение тонического напряжения артериол височных областей мозга и снижение пульсового давления, т. е. являются более утомительными. По окончании теоретической части ДАД увеличилось на 5,8 % ($p < 0,05$), ср. АД — на 3,1 % ($p < 0,05$). Пульсовое давление (ПАД) снижалось незначительно. К концу работы на компьютере снижение ПАД составило 13,0 % ($p < 0,05$), а показатель, характеризующий тонус мелких сосудов мозга, возрос на 4,5 % ($p < 0,05$). У учащихся X класса (4 ч в неделю практических занятий на ЭВМ) по сравнению с рассматриваемыми ранее группами сдвиги показателей системы кровообращения были более выраженными. Происходило снижение частоты сердечбиений (15,2 %, $p < 0,05$), систолического давления (7,1 %, $p < 0,05$), пульсового давления (24,1 %, $p < 0,05$), значительное повышение тонуса мелких сосудов мозга (14,9 %, $p < 0,05$). Такие изменения гемодинамики связаны, видимо, с умственным напряжением в процессе труда. Полученные данные показали, что у юношей 15—17 лет занятия, различные по интенсивности, вызывают, при сходном характере реакции, разную степень изменений сердечно-сосудистой системы. При этом, чем интенсивнее нагрузка, тем более выраженные изменения наблюдались со стороны показателей центрального и мозгового кровообращения.

Обследование учащихся в конце учебного года показало, что изменения параметров системы кровообращения в ответ на умственную нагрузку при работе на ЭВМ были сходны с полученными в I четверти, но ме-

нее выраженными, что свидетельствует, по нашему мнению, о повышении функциональной активности сердечно-сосудистой системы.

Реакции системы кровообращения школьников при работе на ЭВМ сводились к следующему: повышение диастолического давления, снижение пульсового давления, урежение сердечбиений и возрастание тонического напряжения мелких мозговых артерий.

Указанные изменения сердечно-сосудистой системы в течение дня можно расценивать как проявление состояния умственного утомления [2, 3, 8]. Анализ частоты сердечбиений и тонического напряжения мозговых сосудов показал, что достоверные изменения этих показателей регистрируются с 35-й мин занятий, что соответствует результатам исследований, функционального состояния высшей нервной деятельности старшеклассников в условиях дисплейного класса [4, 5].

Наиболее выраженное напряжение функционирования сердечно-сосудистой системы отмечается при занятиях во вторую смену. К 16 ч происходило значительное урежение частоты сердечбиений (ЧСС) — 18 %, $p < 0,05$, снижение пульсового давления (ПАД) — 17 %, $p < 0,05$, выраженное повышение диастолического давления (ДАД) — 10 %, $p < 0,05$, что является отражением внутреннего десинхроноза функций сердечно-сосудистой системы [6, 7].

Выявлена зависимость характера и выраженности гемодинамических отклонений от уровня эмоциональной стабильности испытуемых. У учащихся с высокой эмоциональной устойчивостью сдвиги абсолютных величин показателей центрального и мозгового кровообращения были незначительными. У юношей этой группы хорошо выражены реакции перераспределения мозгового кровообращения. Учащиеся со средней эмоциональной стабильностью характеризуются более выраженной реакцией сердечно-сосудистой системы (снижение САД 8 %, $p < 0,05$; урежение ЧСС 13 %, $p < 0,05$; увеличение тонуса мозговых артериол 26 %, $p < 0,05$), отражающей значительное нервно-эмоциональное напряжение.

Выявлена также зависимость реактивности мозгового и центрального кровообращения юношей от типа мануальной асимметрии. У леворуких юношей происходили существенные изменения как центрального, так и мозгового кровообращения, что отражает малозффективный тип регуляции. У учащихся с правосторонней мануальной асимметрией наблюдались лишь локальные изменения мозговой гемодинамики.

Полученные результаты позволили сделать следующие рекомендации по организации занятий в профильных классах, обеспечивающих подготовку лаборантов-программистов и операторов ЭВМ на базе УПК:

1. Организовать через каждые полчаса работы 10-минутные перерывы.

2. При составлении расписания занятия указанных групп ставить в утренние часы.

3. Отбор в группы профессиональной подготовки для работы на ЭВМ проводить с учетом уровня эмоциональной стабильности и мануальной асимметрии.

Литература

1. Гельтищева Е. А., Селихова Г. Н. Режим работы за дисплеем // Информатика и образование. 1987. № 1. С. 82—84.

2. Калужная Р. А. Физиология и патология сердечно-сосудистой системы детей и подростков. М.: Медицина, 1973.

3. Кан Е. Л., Аветикян Ш. Т., Кан Г. С. Реакции сердечно-сосудистой системы диспетчеров управления воздушным движением в условиях, моделирующих профессиональную деятельность. М., 1987. 5 с. Деп. в ВИНТИ 02.06.87. № 3901-В87.

4. Леонова Л. А., Савватеева С. С. Занятия в КВТ: влияние на организм // Информатика и образование. 1986. № 3.

5. Леонова Л. А. Физиолого-гигиенические основы организации учебных занятий школьников с применением электронно-вычислительной техники // Новые исследования в психологии и возрастной физиологии. 1989. Вып. 1.

6. Сиякина А. Д. Гигиеническая рационализация режима дня школьников Крайнего Севера на основе характеристик циркадного ритма сердечно-сосудистой системы: Автореф. дис. ... канд. биол. наук. М., 1984. 23 с.

7. Функциональные системы организма: Руководство / Под ред. К. В. Судакова. М.: Медицина, 1987.

8. Яруллин Х. Х. Клиническая реоэнцефалография. М.: Медицина, 1983.

60

Лучше меньше, да лучше

Ценность книг не принято измерять их весом. И все же солидные фолианты всегда занимают на полках почетное место.

Величина компьютера также не является показателем его возможностей, однако японская фирма Fujitsu с радостью сообщает, что с ее конвейеров сошла новая модель переносного компьютера Notebook PC размером с книгу, преодолевшая килограммовый барьер.

Несмотря на более чем скромный вес, малютка имеет характеристики стандартного компьютера, которые обеспечивает процессор 80С286, работающий на частоте 4/8 МГц, и память 640 Кбайт. Кроме того, «книжка» имеет два дополнительных слота для расширения объема ОЗУ до 1,6 Мбайт.

В верхнюю часть корпуса очень естественно встроен жидкокристаллический дисплей, который может отобразить 640 двухцветных точек в каждой из 400 горизонтальных линий.

ЧТО? МОЖЕТ? ЭВМ

Все это аппаратное многообразие конструкторы умудрились разместить в коробке толщиной с обычную видеокассету, которую можно полностью накрыть стандартным листом писчей бумаги. Ну а вес 990 г, как уже говорилось, стал особой гордостью разработчиков. И это несмотря на то, что в этом же объеме разместилась стандартная версия довольно большой операционной системы MS-DOS ROM V. 3.22, которая может работать от встроенной в компьютер батареи восемь часов подряд.

Контакт без контакта

Одним из недостатков компактных IC-карт, которые могут служить кредитками, пропуском, хранилищем программ для калькулятора и т. д., является электрический разъем, с помощью

которого карта подключается к соответствующему контроллеру компьютера. Электрический контакт такого разъема боится вибрации, со временем становится менее надежным, а его металлические выводы служат «портом ввода» для разрушительного статического электричества.

От всех этих недостатков избавили свою новую бесконтактную электронную карточку на фирме TDK. Данные, которыми карта обменивается с внешними устройствами, передаются с помощью электромагнитных полей, излучаемых встроенной в карту катушкой. Катушка расположена в центре карты, что позволяет ее владельцу не думать о том, какой стороной вставлять ее в щель адаптера. Кстати, ширина щели не превышает двух с половиной миллиметров, благодаря высокоточной технологии, применяемой при изготовлении карты.

Новая карта выпускается в двух вариантах — на 8 и 16 Кбайт — и поставляется с электромагнитным адаптером, позволяющим использовать новые карты в старых системах.

О «Корвете»

Компьютер — это одновременно и бесконечно сложное устройство, и очень простое.

Бесконечно сложное, потому что его «поведение» как программируемого устройства может быть сколь угодно разнообразно.

Очень простое, потому что его «поведение» однозначно определяется некими правилами, достаточно знать их, чтобы заставить компьютер сделать именно то, что надо.

Правила обращения с разными компьютерами, набор их пользовательских программ и управление этими программами в общем сходны, однако в деталях все же имеются существенные различия, что ставит в тяжелое положение обладателей ПЭВМ, не имеющих адекватных описаний своих машин.

В «ИНФО» начинается публикация цикла статей, предназначенных сделать более простым общение с «Корветом», помочь ориентироваться в море имеющегося программного обеспечения, дать желающим возможность постигнуть самые глубинные тайны этой ПЭВМ — такие, как работа с графикой, дисками и т. д. на аппаратном уровне. Начиная с этого номера печатаются статьи, описывающие архитектуру «Корвета» и его процессора. Предполагается публикация материалов по программированию БИС, входящих в состав «Корвета». Особое внимание будет уделено работе с графикой, дисками и прерываниям.

При описании порта расширения помимо тонкостей программирования будет рассказано о том, как подключить к «Корвету» различные нестандартные устройства (робот, вольтметр и др.).

Программному обеспечению будут посвящены обзоры-путеводители по функционально схожим прикладным пакетам, где будет рассказано о редакторах текста (чем они различаются, достоинства и недостатки различных версий), программах для работы с принтером (как распечатать подготовленные документы), операционных системах (включая сетевые версии), графических и музыкальных редакторах и многом другом.

Наиболее интересные и важные пакеты будут представлены «солю», чтобы облегчить их освоение и позволить читателям узнать все тонкости их применения. В первую очередь будет описана интегрированная сре-

да Express Pascal (аналог широко распространенной системы Turbo Pascal версии 4.0 фирмы Borland), специально написанная для «Корвета». Она включает редактор текста и компилятор. Высокая скорость работы, возможность работы редактора и компилятора и запуска результирующей программы без обращений к диску, богатый набор встроенных процедур и функций дают уникальные возможности при обучении школьников алгоритмическому языку высокого уровня. Кроме того, компилятор генерирует достаточно компактный и эффективный код, так что Express Pascal может успешно применяться для написания прикладных пакетов. В частности, на нем написана инструментальная среда для создания обучающих программ «Радуга».

Описание «Радуги» также готовится к публикации. Использование специального инструмента для создания обучающих программ позволяет значительно сократить затрачиваемое на это время. Более того, на основе такой системы можно в короткое время разработать обучающую программу, даже не имея навыков программирования. Эта особенность «Радуги» позволяет вовлечь в процесс создания обучающих программ преподавателей-предметников. В частности, каждый учитель может реализовать собственные представления о том, как преподнести те или иные темы, создавать собственные варианты контрольных работ для учеников и многое другое.

Важность создания и использования обучающих программ по предметам, не связанным с информатикой (по физике, химии, математике, русскому и иностранному языкам и т. д., может быть, даже по физкультуре) трудно переоценить. Возможности компьютера (графика, динамика экрана, многовариантность реакции на поведение обучаемого) позволяют в корне изменить технологию обучения, реализовать методические идеи, невозможные в книжном варианте, и т. д. Кроме того, способность компьютера к моделированию реальных процессов позволяет сделать обучение более наглядным, использование компьютерных лабораторных работ углубит знание предмета (последнее особенно важно, поскольку диапазон лабораторных работ в школе крайне ограничен из-за

высокой стоимости оборудования и опасности многих экспериментов).

В настоящее время имеются отдельные обучающие программы по различным предметам, однако пакета, охватывающего целый курс, нет (по крайней мере у нас нет сведений об этом). Создание такого пакета — дело дорогостоящее и длительное, требует привлечения большого числа специалистов. Однако многие организации ведут работы в этой области. В частности, физический факультет МГУ, НИИЯФ МГУ и МП «Микс» при НИИЯФ МГУ совместно создают курс обучения физике для средних школ. Общий объем курса составит 80—90 ч. В него включены разделы «Механика» (25 ч), «Молекулярная физика и теплота» (15 ч), «Электричество и магнетизм» (30 ч), «Оптика» (10 ч), «Физика микромира» (15 ч). В полном объеме курс предполагается завершить к 1993 г., однако отдельные его части будут распространяться уже в начале 1992 г.

Важной его особенностью будет наличие сопровождающего компьютерного физического практикума. Кроме реализации на компьютере достаточно традиционных лабораторных работ в практикуме будут изучаться явления, не реализуемые в природе, а также те, которые невозможно наблюдать в лабораторных условиях.

Некоторые материалы по «Корвету» (очень важные и нужные, но не всем, а также не вписывающиеся в тематику «ИНФО») будут выходить отдельными изданиями. Однако в журнале будут печататься их аннотации, и желающие смогут (мы надеемся) заказать их. Так будут опубликованы, например, учебник по Паскалю, техническое описание «Корвета» и т. д.

Среди авторов статей будут разработчики «Корвета» и оборудования к нему, разработчики программного обеспечения, преподаватели вузов и школ.

Заявленные выше материалы будут печататься примерно в течение года. А что же делать тем, кто уже сейчас испытывает трудности в выборе и приобретении различного программного обеспечения, у кого возникли неразрешимые проблемы? В этом случае вы можете обратиться в МП «Микс» по адресу: 119899, Москва, ГСП, Ленинские горы, МГУ, НИИЯФ, МП «Микс»; тел. 939-49-49.

Одна из целей деятельности этого малого предприятия — разработка и сопровождение школьного программного обеспечения и оборудования для ПЭВМ «Корвет». Здесь вы можете получить самые свежие сведения о разработках нового программного обеспечения для «Корвета» и о новых версиях уже существующего, обновить и улучшить имеющийся у вас комплект программ. В на-

стоящее время «Микс» предлагает:
пакет программ ОС CP/M-80;
дискетный редактор;
АБРИС — графический редактор;
музыкальный редактор;
пакет для работы с принтером;
пакет языка Си с графической библиотекой;

пакет языка PASCAL MT+ с дополнительными библиотеками;
пакет программ языка ассемблер;
SUPERTEXT — текстовый редактор;
ChiWorker — аналог широкого распространения среди пользователей IBM PC текстового редактора ChiWriter, позволяет использовать несколько наборов символов (в том числе математические и т. п.) одновременно и создавать «нелинейные» тексты (сложные математические и химические формулы и т. п.);

KOPNET — сетевая операционная система для Корвета;

XAPC — архиватор файлов;
EXPRESS PASCAL — интегрированная среда для работы с Паскалем;

VCC — словарь английского языка на 20 000 слов;

E — редактор текста для работы в локальной сети KOPNET;

РАДУГА — инструментальная среда для написания обучающих программ;
пакеты игровых программ.

В заключение — несколько советов по выбору и приобретению программного обеспечения.

Во-первых, не приобретайте программное обеспечение у случайных поставщиков, особенно если вы не уверены, что программы поставляются на законных основаниях. Несанкционированно проданное программное обеспечение не сопровождается, т. е. в случае обнаружения ошибок в приобретенной программе вам не дадут исправленную версию; если у вас появились вопросы по работе с программой, вам не ответят.

Во-вторых, обращайтесь внимание на знаки авторства. Крупных разработчиков не так много, и вы скоро поймете, чьи программы имеет смысл приобретать, а от чьих лучше воздержаться. Необходима осторожность с программами, авторство которых не ясно.

В качестве «авторитетных» можно отметить следующие знаки: mkSoft; «кошка»; Микс; НИИЯФ МГУ; WOODMAN.

Ну, и последнее — призыв: пишите, задавайте вопросы, вносите предложения. Это поможет нам выбрать нужные темы для обсуждения и скорректировать точку зрения.

С. АХМАНОВ, М. КОВТУН,
А. НЕЧАЕВ, Н. РОЙ, А. СКУРИХИН
НИИЯФ МГУ

Архитектура «Корвета»

«Корвет», как и любой другой компьютер, состоит из процессора, памяти и устройств ввода-вывода (УВВ), называемых еще периферией. С аппаратной точки зрения компьютеры отличаются друг от друга типом процессора, объемом и типом памяти, составом УВВ и способом доступа к ним, быстродействием и т. п. Дополнительные отличия, весьма заметные для пользователя, может вносить имеющееся программное обеспечение (тип операционной системы и др.).

Мы постараемся дать детальное представление о том, как устроен «Корвет» для программиста нижнего уровня, напрямую управляющего аппаратурой. Программного обеспечения будем касаться лишь вскользь, и то главным образом встроенного.

Основой «Корвета» является 8-разрядный микропроцессор КР580ВМ80А (подробнее о нем рассказывается в статье «Архитектура процессора КР580ВМ80А»).

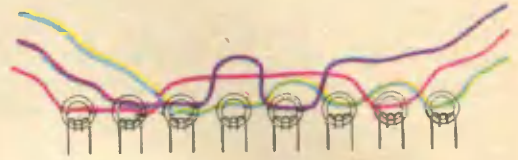
Процессор общается с другими устройствами (памятью и УВВ) посредством чтения и записи чисел, в зависимости от контекста называемых данными или командами. Память — это массив пронумерованных ячеек, предназначенных для хранения данных или команд. Память бывает постоянной (постоянное запоминающее устройство, ПЗУ) и оперативной (оперативное запоминающее устройство, ОЗУ).

ПЗУ — важнейший компонент компьютера. Компьютер без программы — беспомощная железка, но сразу после его включения, когда ОЗУ еще младенчески чисто, запускается программа, хранящаяся в ПЗУ. Как правило, она производит проверку основных узлов компьютера и загружает операционную систему.

Может ли существовать процессор столь «умный», чтобы самостоятельно загрузить какую-либо программу из внешней памяти? Да, такие процессоры существуют, но процедура загрузки в них все равно записана в ПЗУ, только внутреннее, являющееся частью процессора (так называемое ПЗУ микропрограмм).

Способы реализации ПЗУ весьма многочисленны. В доисторические времена, когда компьютеры изготавливали из транзисторов и даже радиоламп, вся внутренняя память делалась на ферритовых кольцах; часть «ферритового» ПЗУ изображена на рис. 1. Количество колец в одном ряду обычно равно разрядности ячейки памяти, в нашем случае 8. Каждое кольцо имеет обмотку считывания, кроме того, через кольца продеты провода, образующие собственно ячейки памяти — каждый провод свою ячейку. Если какой-то бит в данной ячейке должен быть равен 1, провод проходит сквозь кольцо, если 0 — мимо. Благодаря этому

при подаче импульса тока на провод в обмотках считывания некоторых колец возбуждаются индуцированные токи, а в некоторых — нет; происходит считывание информации.



Отсюда видно, что процесс записи в такое устройство заключается в проделывании проводов через определенные кольца, весьма напоминающем процесс шитья; отсюда и происходит термин «прошивка ПЗУ», сохранившийся до сих пор, хотя сейчас программирование постоянной памяти делается совсем иначе.

Если ПЗУ нужно изготовить в большом числе экземпляров, используют так называемые масочные микросхемы. Их содержимое определяется в процессе конструирования и не может быть изменено, зато они наиболее дешевы.

В программируемые ПЗУ (ППЗУ) информация заносится после их изготовления. В ППЗУ с плавкими перемычками (это, например, микросхемы типа 556РТ) просто пережигают импульсами тока некоторые перемычки, записывая таким образом нули в некоторые биты. Перепрограммировать их после этого нельзя (разве что добавить нулей), они потребляют много энергии, однако имеют высокое быстродействие.

Существуют и такие ППЗУ, которые можно программировать неоднократно, стирая старую информацию — в некоторых ультрафиолетовым светом, направляемым через специальное окошко в корпусе (это микросхемы типа 573РФ), в некоторых — электрическим током (например, 558РР; в подобных микросхемах для считывания используются напряжения порядка 5 В, для стирания — 24 В, для программирования — 15—20 В).

ПЗУ содержит данные, записываемые обычно на заводе при изготовлении компьютера. Программист не может изменить содержимое ПЗУ; более того, изменение содержимого ПЗУ в процессе эксплуатации является неисправностью компьютера и диагностируется тестом начального пуска — в этом случае выдается сообщение «ПЗУ неисправно».

Объем (количество ячеек) ПЗУ равен 24 Кбайт. Первые 8 Кбайт содержат тест начального пуска, программу инициализации устройств и переменных, драйверы дисплея,

клавиатуры, локальной сети и некоторых других устройств, загрузчики с диска и из внешнего ПЗУ. Оставшиеся 16 Кбайт содержат интерпретатор языка Бейсик, который включается, если «Корвет» не может загрузить операционную систему с диска. Подробное описание ПЗУ «Корвета» будет дано позже.

ОЗУ позволяет не только считывать данные из своих ячеек, но и записывать их туда. Именно в ОЗУ загружаются операционная система и программы, которые вы запускаете с диска, локальной сети или магнитофона. Здесь хранятся также все переменные и данные, которые используют программы.

Общий объем ОЗУ в «Корвете» — 64 Кбайт. Этот массив ячеек называют еще основной памятью.



Другой тип устройств, к которым может обращаться процессор — УВВ. Их набор в «Корвете» достаточно широк: алфавитно-цифровой и графический дисплеи, клавиатура, интерфейсы принтера, магнитофона, многофункциональный параллельный, а также контроллеры локальной сети, накопителя на гибких дисках, последовательного интерфейса.

Дисплеи и клавиатура имеют в своем составе довольно большие массивы ячеек. В состав алфавитно-цифрового дисплея входят 1024 9-битовые ячейки, графического — до 192 Кбайт (в школьном варианте обычно используется версия 48 Кбайт), клавиатура включает 512 ячеек (отметим, что ячейки клавиатуры можно только читать).

Остальные устройства управляются через отдельные ячейки, называемые управляющими и периферийными регистрами (не путать с регистрами процессора!). Отметим, что оба дисплея имеют помимо массивов ячеек управляющие регистры.

Регистры устройств в «Корвете» бывают двух типов: доступные и для чтения, и для записи (периферийные, принадлежат дисководу, параллельному и последовательному интерфейсам и т. д.); доступные только для записи (управляющие — к их числу относятся системные и некоторые регистры графического контроллера). Регистров каж-

дого типа — 256, т. е. всего они занимают 512 ячеек.

Если просуммировать все перечисленные ячейки, то получается, что процессор должен различать (адресовать) до 282 Кбайт, а он умеет адресовать только 64 К. Где же выход из этой ситуации?

Обычно для расширения адресного пространства процессора используется так называемая страничная организация памяти, например, с помощью введения сегментного регистра. 20-разрядный адрес (т. е. возможность адресовать 1 Мбайт) с его помощью получается так: к 16 разрядам адреса добавляются 4 разряда смещения из сегментного регистра (рис. 2). Другими словами, 1 Мбайт памяти составляется из 16 «страниц» по 64 Кбайт; номер страницы задается в

сегментном регистре, а внутри страницы ячейки адресуются непосредственно микропроцессором. Такая схема используется, в частности, в компьютерах IBM PC.

В «Корвете» используется другой способ «ужимания» большого числа ячеек в небольшое адресное пространство процессора. Исходно все адресное пространство занимает ОЗУ (основная память). Все остальные ячейки можно включать в адресное пространство и отключать от него.

Если ячейки какого-то устройства (далее будем называть их просто устройством) включаются в адресное пространство, то соответствующая область ОЗУ отключается, т. е. устройство как бы заслоняет ОЗУ от процессора. Но устройства, в которые можно только писать или из которых можно только читать, лишь частично заслоняют ОЗУ («под» ячейку «только для чтения» можно записать в ОЗУ данные, «из-под» ячейки «только для записи» данные можно считывать). В связи с этим уместно предостеречь программистов, любящих скидывать ненужные константы в нейтральную, как они считают, область, например в ПЗУ. В «Корвете» эта константа запишется в соответствующую ячейку ОЗУ, даже если ее сверху «накрывает» ПЗУ.

Произвольно подключать и отключать отдельные устройства нельзя — используются только некоторые наборы из всего возмож-

ного множества комбинаций. Каждый такой набор называется конфигурацией адресного пространства, а расположение устройств в адресном пространстве — картой памяти. На «Корвете» можно использовать 32 конфигурации адресного пространства.

Для задания той или иной конфигурации адресного пространства в состав системы управления памятью «Корвета» входит системный управляющий регистр. Он наиболее важен и требует очень осторожного обращения: при изменении его содержимого мгновенно меняется карта памяти (представьте, что почувствует турист, заметив, что кто-то подменил карту местности, по которой он путешествует!). Если это произошло в результате необдуманных действий, то машина зависает и приходится делать холодный старт. При этом возможна порча информации, записанной на диски.

В системном регистре для задания номера конфигурации используются биты со 2-го по 6-й (5 битов как раз позволяют задать 32 конфигурации); остальные должны быть установлены в 0.

Часть конфигураций отвечает картам памяти, используемым в известных операционных системах CP/M, TRS 80, МикроДос и др. Отметим, что не все конфигурации памяти используются, а некоторые вовсе не имеют смысла — в них можно войти, но из них нельзя выйти. Появление таких бессмысленных конфигураций объясняется тем, что биты, определяющие номер конфигурации, разбиты на отдельные группы, каждая из которых отвечает на свои системы, и в некоторых комбинациях они оказываются не согласованы.

Возможные карты памяти (области памяти, отводимые под различные устройства в зависимости от кода, записанного в системном регистре) представлены в таблице. Для алфавитно-цифрового дисплея (А/Ц), графического дисплея (ГЗУ), периферийных (для чтения-записи) регистров (Per1), управляющих (только для записи) регистров (Per2) и клавиатуры приведены только начальные адреса их областей, поскольку длины областей фиксированы: для А/Ц — 3FF, ГЗУ — 3FFF, регистров — по FF, клавиатуры — 1FF.

Конфигурация 0 соответствует карте памяти, принятой в машинах семейства TRS 80 фирмы Tandy. Конфигурация 1CH соответствует карте памяти, используемой в ОС CP/M. МикроДос, в зависимости от версии, использует конфигурацию 5CH или 1CH. Бейсик, зашитый в ПЗУ, использует конфигурацию 40H.

При изменении карты памяти неприятные

Код	ПЗУ	А/Ц	ОЗУ	ГЗУ	Per1	Per2	Клав.
00	0-37FF	3C00	4000-FFFF		3800	3A00	3B00
04	0-1FFF		2000-FFFF				
08	0-3FFF		4000-FFFF				
0C			0000-FFFF				
10	0-1FFF	FC00	2000-F7FF		F800	FA00	F800
14	0-1FFF	FC00	2000-F7FF		F800	FA00	F800
18	0-3FFF	FC00	4000-F7FF		F800	FA00	F800
1C		FC00	0000-F7FF		F800	FA00	F800
20	0-37FF	3C00	4000-BFFF	C000	3B00	3A00	3B00
24	0-1FFF		2000-BFFF	C000			
28	0-1FFF		4000-BFFF	C000			
2C			0000-BFFF	C000			
30	0-1FFF		2000-3FFF	4000	FE00	FF00	
34	0-1FFF		0000-3FFF	4000	FE00	FF00	
38	0-3FFF		0000-FDFF	4000	FE00	FF00	
3C			0000-3FFF	4000	FE00	FF00	
			0000-FDFF				
40	0-3FFF	FC00	4000-F7FF		F800	FA00	F800
44	0-1FFF	FC00	2000-F7FF		F800	FA00	F800
48	0-3FFF	FC00	4000-F7FF		F800	FA00	F800
4C		FC00	0000-F7FF		F800	FA00	F800
50	0-3FFF		4000-FDFF	FE00	FF00		
54	0-1FFF		2000-FDFF	FE00	FF00		
58	0-3FFF		4000-FDFF	FE00	FF00		
5C			0000-FDFF	FE00	FF00		
60	0-3FFF		6000-BEFF	C000		BF00	
64	0-1FFF		2000-BEFF	C000		BF00	
68	0-3FFF		4000-BEFF	C000		BF00	
6C			0000-BEFF	C000		BF00	
70	0-3FFF		6000-BFFF	C000			
74	0-1FFF		2000-BFFF	C000			
78	0-3FFF		4000-BFFF	C000			
7C			0000-BFFF	C000			

сюрпризы чаще всего преподносят прерывания. Они происходят «сами по себе», независимо от наших действий, после чего управление передается туда, где должна находиться программа обработки прерывания. А если мы перед этим изменили карту памяти, этой программы на привычном месте, скорее всего, не окажется. Результат — зависание. Следовательно, при переключении карты памяти надо запретить прерывания (с помощью команды DI).

Прочитать содержимое системного регистра нельзя, следовательно, все манипуляции с картой памяти должны задаваться программистом заранее и быть хорошо продуманными. Нужные части программы не должны оказаться в «отключенной» части памяти. Если программа начинает работать под управлением CP/M, в системном регистре обычно записано число 1CH; поэтому по окончании манипуляций с картой памяти следует (если нужно вернуть управление CP/M) загрузить в системный регистр 1CH и разрешить прерывания (с помощью команды EI).

Системный регистр имеет адрес 7FH на странице управляющих регистров. Это значит, что при использовании карты памяти TRS 80 его абсолютный адрес равен 3A7FH. Тот же адрес в конфигурации ОС CP/M равен FA7FH. Таким образом, если мы хотим изменить карту памяти из CP/M в TRS 80 и обратно, нам потребуется следующая программа:

SRESC EQU	0FA7FH	; Адрес системного
		; регистра в CP/M
SREGT EQU	3A7FH	; Адрес системного
		; регистра в TRS80
ORG	4000H	; Начало ОЗУ TRS80
DI		; Запрещение
		; прерываний
MVI	A, 0	; Номер конфигурации
		; TRS80
STA	SREGC	; Запись нового
		; значения
		; системного
		; регистра
		; Часть программы,
		; работающая с
		; картой памяти
		; TRS80
MVI	A, 1CH	; Номер конфигурации
		; CP/M
STA	SREGT	; Запись нового
		; значения
		; системного
		; регистра
EI		; Разрешение
		; прерываний

66

Здесь специально подчеркнуто, что программа, переключающая карту памяти, должна начинаться с адреса, большего или равного 4000H, поскольку в конфигурации TRS 80 именно с этого адреса начинается ОЗУ.

Переключив карту памяти, нельзя пользоваться никакими функциями BIOS и BDOS. По этой причине не пытайтесь переключать карту памяти из Бейсика — компьютер наверняка зависнет. Программа, занимающаяся этим опасным делом, должна создаваться

при полном осознании происходящих в машине процессов. Если вы не знаете ассемблера, лучше не трогайте карту памяти, тем более что ее изменения нужны лишь в небольшом числе случаев — обычно при работе с графикой и «изъятии» данных из ПЗУ. В языках программирования есть специальные библиотеки для выполнения этих функций — пользуйтесь ими.

Как видно из таблицы конфигураций памяти, ГЗУ может занимать окно размером 16 Кбайт. Позднее мы увидим, что это фундаментальная единица объема графической памяти, определяющая цветовую плоскость. Для использования ГЗУ необходимо переключить карту памяти таким образом, чтобы 16-Кбайтное окно оказалось подключенным к адресному пространству процессора. При этом желательно минимально повредить нашу программу, т. е. сохранить в адресном пространстве те области памяти, где записаны необходимые ее части. Например, если мы работаем под управлением ОС CP/M, т. е. в системном регистре хранится 1CH, то для подключения ГЗУ можно использовать конфигурацию 6CH: ГЗУ будет подключено к верхним 16 Кбайт памяти, нашей программе станут недоступными функции BDOS, BIOS и периферийные регистры. При работе с конфигурацией 40H, соответствующей Бейсику без графики, удобно подключить графику, используя конфигурацию 60H. Именно так поступает Бейсик, но не пытайтесь сделать это из Бейсика сами!

Продолжение следует.

С. АХМАНОВ, А. НЕЧАЕВ, А. СКУРИХИН

Архитектура процессора KP580BM80A

Мы попытаемся рассказать, что такое процессор KP580BM80A, используемый в ПЭВМ «Корвет», с точки зрения программиста, и надеемся, что, прочитав статью, вы сможете начать писать программы на ассемблере.

Прежде чем перейти к описанию процессора, поговорим немного о том, что такое компьютер, из чего он состоит, какое место в нем занимает процессор.

Компьютер — довольно сложно организованное устройство, способное к активным действиям; предполагается, что они должны приносить определенную пользу. Польза может быть самая разнообразная — вычислить

значение какой-либо функции, занять хозяина компьютерной игрой и т. д.

Наверно, существует много способов заставить сложно организованную систему действовать таким образом, чтобы достичь какой-то конкретной цели, однако человечество открыло всего два. При первом каждый компонент системы стремится к достижению собственных целей; отдельные компоненты имеют возможность «договариваться» (изменять свое поведение в результате взаимодействия) друг с другом. Если цели отдельных компонентов не слишком различны, все они могут быть достигнуты совместно, и есть

шанс, что система в целом будет двигаться к какой-то общей цели.

Примерно так функционирует свободное общество людей; примерно так работают некоторые из так называемых нейрокомпьютеров.

Если же цели системы и компонентов сильно различаются или даже противоречат друг другу, то модель свободного общества непригодна для достижения этих целей. Поэтому человечество изобрело второй способ организации — централизованное управление.

Такую систему не интересуют цели отдельных компонентов (людей). Единственное, что безусловно требуется, — исполнение распоряжений. Компонент, не выполняющий распоряжений, считается неисправным и устраняется.

Цель централизованной системы не может возникнуть естественным образом, как равнодействующая целей компонентов. Общей целью может быть провозглашена цель одного элемента системы (автократия) или группы элементов (демократия); цель может быть привнесена и извне (как, например, в случае колонизированной страны).

Компьютер — система с централизованным управлением. Это обусловлено как тем, что отдельные компоненты компьютера не имеют собственных целей, так и тем, что цель компьютеру задается извне — он должен делать то, что хотите вы.

Когда вы что-то захотели от компьютера и сформулировали свое желание в виде достижимой для него цели, в дело вступает программист. Его аналогом в человеческом обществе можно считать подсистему, обычно именуемую *аппаратом* (ту самую, которую все ругают. Впрочем, программистам тоже достается). К сожалению, в нашем обществе деятельность этой подсистемы скрыта, хотя именно она является ключевой: какие бы ни были провозглашены цели, система будет делать то, что задаст аппарат (или программист).

К счастью, программисту обычно не удается замаскировать свою деятельность: всегда можно быстро сравнить «продукцию» компьютера с поставленными целями и, если результат сравнения вас не устраивает, отказать от услуг данного программиста. С аппаратом это сделать гораздо сложнее — не вы, а аппарат может отказаться от ваших услуг, если он вас не устраивает.

После того как программист уяснил, чего же вы хотите, он придумывает способ (алгоритм) достижения вашей цели, а затем трансформирует его в пакет распоряжений для компьютера. Этот пакет распоряжений обычно называют программой, а отдельные распо-

ряжения — командами.

С программой работает уже сам компьютер.

Он состоит из трех подсистем — процессора, памяти и устройств ввода-вывода (УВВ). Подсистема УВВ связывает компьютер с внешним миром. Без нее компьютер не может ничего увидеть, ничего услышать, ничего сказать. В ее состав обычно входят клавиатура, дисплей, принтер, дисковод, мышь и т. д. Чтобы передать компьютеру программу, необходимо воспользоваться подсистемой УВВ.

Хранится программа в оперативной памяти (ОЗУ). Казалось бы, ОЗУ не является обязательным компонентом компьютера: можно не передавать компьютеру весь пакет распоряжений сразу, а давать по одному — пусть сначала выполнит одну команду, потом следующую и т. д. Так сделать действительно можно. Однако отдельные команды выполняются очень быстро, а подсистема ввода-вывода работает очень медленно (в сотни и тысячи раз медленнее). Поэтому отсутствие ОЗУ лишает компьютер его основного достоинства — способности очень быстро делать различные операции.

Кроме программного счетчика имеется еще один 16-разрядный регистр, предназначенный для хранения адреса — SP (от Stack Pointer — указатель стека). Стек — это особый способ хранения данных, устроенный по образу и подобию пистолетной обоймы (думается, что стараниями военруков даже девочки в нашей стране хорошо представляют себе, что это такое). Отличие только в том, что патроны утапливаются в обойму, а данные «кладутся» на стек сверху, после чего его «стенки» «подрастают». На «верхнее» (всегда на очередное «верхнее») данное и указывает содержимое SP. Такой механизм хранения данных (LIFO — last in first out, последним вошел — первым вышел) удобен для реализации некоторых алгоритмов, в частности при использовании подпрограмм.

Регистры A, B, C, D, E, H, L — 8-разрядные, предназначены в основном для хранения и обработки данных. Самый важный из них — A (его называют аккумулятором). С его участием выполняются практически все операции над данными (сложение, вычитание, сдвиг, логические операции и т. д.). Результат операции остается в аккумуляторе. Если какая-либо операция предполагает два операнда, то в качестве второго обычно используется содержимое одного из семи описываемых регистров.

С содержимым регистров B, C, D, E, H, L можно выполнять только две арифметические операции — увеличение и уменьшение содержимого на единицу. Зато эти регистры

могут объединяться в пары, образуя три 16-разрядных регистра: BC, DE и HL. Эти «двойные» регистры имеют свои специализации. Основная их функция — косвенная адресация (так называют использование ячейки, адрес которой не задан в команде явно, а записан в заданном в команде регистре). Команды с косвенной адресацией, использующие BC и DE, позволяют производить только обмен данными между аккумулятором и памятью; HL в качестве адресного регистра используют значительно более разнообразные команды.

С 16-разрядными регистрами BC, DE, HL и SP возможны и некоторые арифметические операции: увеличение и уменьшение содержимого на 1, а также сложение. Операция сложения требует двух операндов. Один из них всегда регистр HL, его иногда называют 16-разрядным аккумулятором. Другой — один из BC, DE, SP или снова HL.

Особняком стоит регистр флагов (его еще называют регистром состояния). Он «скрыт» от программиста — нельзя что-то записать в него так же просто, как это можно сделать с другими регистрами.

Хотя современные компьютеры имеют довольно большой объем памяти, бывают случаи, когда вся программа не может поместиться в ОЗУ целиком. В этом случае ее дробят на отдельные части (их называют оверлейными модулями), которые по мере необходимости «подгружаются» в память из УВВ.

Процессор выполняет распоряжения и управляет всеми остальными подсистемами. Чтобы вполне понять, как это происходит, необходимо ознакомиться хотя бы с основными операторами языка ассемблера и элементами архитектуры процессора; в общих же чертах процедуру выполнения команд можно описать так.

Программа загружается в последовательные ячейки памяти (это тоже делает процессор, но как именно, мы пока разбирать не будем — предположим, что это свершается «само собой»), после чего в специальный указатель команды, подлежащей исполнению, имеющийся в процессоре, записывается номер ячейки ОЗУ, содержащей первую команду программы. Далее вступает в действие логика электронной схемы микропроцессора, заданная его конструкцией: команда считывается, анализируется, исполняется, и указатель команды, подлежащей исполнению, увеличивается на 1, заставляя тем самым процессор заняться следующей командой.

А теперь мы можем перейти к описанию функций процессора и их использования. Очевидно, что процессор может выполнить

не любое распоряжение: например, его нельзя послать за пивом. Существует вполне определенный набор распоряжений, которые можно дать процессору. Этот набор называется системой команд. Каждый процессор, в том числе и тот, который мы рассматриваем, имеет свою собственную систему команд.

Хотя разные процессоры обычно имеют отличные системы команд, существуют процессоры, система команд которых совместима с системой команд менее совершенных процессоров. Обычно разработчики обеспечивают это, чтобы обладатели их изделия могли воспользоваться наработанным ранее программным обеспечением. «Старший брат» нашего процессора — Z80 фирмы Zilog; целый ряд совместимых сверху вниз процессоров (т. е. таких, что более совершенные процессоры могут выполнить все команды менее совершенных) выпускает фирма Intel — i8088, i8086, i80286, i80386, i80486.

Но даже если процессоры несовместимы, в их системах команд обязательно присутствуют следующие четыре группы:

команды пересылки (они позволяют переносить содержимое ячеек памяти);

команды операций над содержимым ячеек памяти (позволяют изменять содержимое ячеек в соответствии с некоторыми алгоритмами);

команды управления работой программы; команды управления работой процессора.

Процессор имеет довольно сложную внутреннюю структуру. С точки зрения программиста он состоит из нескольких ячеек памяти, называемых регистрами процессора, и триггеров, состоянием которых можно управлять из программы. Ячейки памяти процессора (далее мы будем называть их регистрами процессора) отличаются от ячеек основной памяти тем, что они не пронумерованы. Вместо номера каждой ячейке присвоено собственное имя — идентификатор. Это подчеркивает, что регистры процессора не равнозначны между собой, более того, они могут иметь разную разрядность (размер). Некоторые из них используются только для хранения и обработки данных, другие используются только для хранения адреса, третьи могут использоваться и как адресные, и для данных.

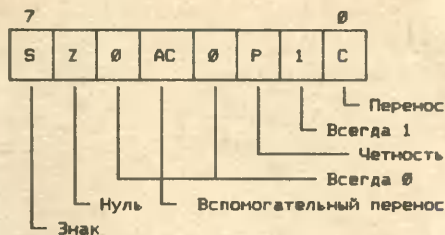
Большинство действий процессора так или иначе связаны с перемещением или изменением содержимого регистров (они изображены на рис. 1).

A	Flags
B	C
D	E
H	L
	SP
	PC

С одним регистром мы уже познакомились под именем указателя команды — это РС (от Program Counter — программный счетчик). Он является 16-разрядным, хранит адрес ячейки, содержащей следующую команду, которую процессор должен выполнить. Когда процессор извлекает из памяти очередное распоряжение, содержимое РС автоматически увеличивается на 1; таким образом процессор готовится извлечь из памяти следующую команду. При подаче на процессор сигнала СБРОС (его нельзя подать с помощью программы, обычно для этого служит специальная кнопка) или при включении питания содержимое РС всегда устанавливается в 0, что обеспечивает определенность состояния процессора при первоначальном запуске (включении питания).

Некоторые команды могут «напрямую» изменять содержимое РС, что бывает необходимо при выполнении условных и безусловных переходов. Позднее мы их внимательно изучим. Все они относятся к группе команд управления работой программы.

Отдельные биты регистра флагов независимы друг от друга. Они устанавливаются в 1 или сбрасываются в 0 в соответствии с результатами некоторых операций процессора (арифметических, логических и др.). Назначение битов регистра флагов показано на рис. 2.



Прочитать «в лоб» регистр флагов также нельзя — о состоянии его битов можно узнать только косвенно, по результатам работы команд условных переходов (о них мы расскажем чуть позже).

Регистр флагов при операциях со стеком образует с аккумулятором 16-разрядное слово, причем содержимое аккумулятора становится старшим, а регистр флагов — младшим байтом этого слова. Эта пара обозначается PSW (Processor State Word — слово состояния процессора). Благодаря этому записать и считать регистр флагов в явном виде все-таки можно, используя стек (подробнее об этом будет рассказано позже).

Кроме описанных выше регистров в процессоре имеется еще одно устройство, доступное программисту, — триггер разрешения прерывания. Работа с прерываниями имеет очень большое значение, поэтому остановимся немного на этой теме.

Продолжение следует

«Корвет»-информ

Техническое описание «Корвета» чересчур объемно для публикации в «ИНФО». Мы приводим лишь информацию о наиболее важных разъемах. Полное техническое описание ПЭВМ «Корвет» со всеми схемами готовится к изданию (ориентировочная цена — 60 рублей). Заказать его можно уже сейчас в МП «Микс».

Адаптер последовательного интерфейса (имеется только в учительской ПЭВМ) предназначен для подключения дополнительных внешних устройств, а в случае использования ПЭВМ в качестве графического терминала — для связи со старшей ЭВМ.

Адаптер построен на базе микросхемы КР580ВВ51А и реализует либо интерфейс типа токовой петли 20 мА (ИРПС), либо интер-

Сигналы на контактах розетки монохромного видеоконтрольного устройства

Номер контакта	Наименование сигнала	Направление	Описание сигнала
01	VIDCOMP	Выход	Смешанный сигнал ВИДЕО и СИНХРО для контроля адвалантно-цифрового контроллера
02	Земля	-	-
03	VIDGR	Выход	Смешанный сигнал ВИДЕО и СИНХРО для подключения монохромного ВКУ
04	Звук	Выход	Канал В программируемого таймера, формирует звуковые колебания программируемого тона
05	-	-	Не используется

Сигналы на контактах разъема расширения

Номер контакта	Наименование сигнала	Направление	Описание сигнала
01 02 03	I/OB02 I/OB01 I/OB00	Вход/Выход	Порт В для подключения адаптеров внешних устройств (разряды 0,1,2)
04 05 06 07 08 09 10 11	I/OCD3 I/OCD2 I/OCD1 I/OCD0 I/OCD4 I/OCD5 I/OCD6 I/OCD7	Вход/Выход	Порт С для подключения адаптеров внешних устройств (разряды 0-7)
12 13 14	Земля Земля T0	- - Выход	- - Канал 0 программируемого таймера
15	2 МГц	Выход	Сигнал частотой 2 МГц
16 17 18 19	I/OA07 I/OA06 I/OA05 I/OA04	Вход/Выход	Порт А для подключения адаптеров внешних устройств (разряды 4-7)
20 21 22 23 24	I/OB03 I/OB04 I/OB05 I/OB06 I/OB07	Вход/Выход	Порт В для подключения адаптеров внешних устройств (разряды 3-7)
25 26 27 28 29 30 31 32	+5 В +5 В +5 В +12 В -12 В Земля Земля C7	- - - - - - - -	- - - - - - - Разряд 7 порта С микросхемы B16
33	IR00	Вход	Вход прерывания с порта расширения
34 35 36 37	I/DAD0 I/DAD1 I/DAD2 I/DAD3	Вход/Выход	Порт А для подключения адаптеров внешних устройств (разряды 0-3)

Сигналы на контактах розетки цветного видеоконтрольного устройства

Номер контакта	Наименование сигнала	Направление	Описание сигнала
01 02	Земля Звук	- Выход	- Канал 0 программируемого таймера, формирует звуковые сигналы программируемого тона
03	VIDR	Выход	Видеосигнал "красный"
04	VIDG	Выход	Видеосигнал "зеленый"
05	VIDB	Выход	Видеосигнал "синий"
06	VIDI	Выход	Видеосигнал "интенсивность"
07	VIDR	Выход	Смешанный сигнал ВИДЕО и СИНХРО для монохромного ВКУ
08 09	HVSYNCR HVSYNCS	Выход Выход	Строчно-кадровый синхронизмус Строчно-кадровый синхронизмус

Сигналы на контактах розетки разъема печатающего устройства

Номер контакта	Наименование сигнала	Направление	Описание сигнала
01	STBUS	Выход	Сигнал стробирования для вывода данных на печать. Активный уровень - низкий
02 03 04 05 06 07 08 09	PD0 PD1 PD2 PD3 PD4 PD5 PD6 PD7	Выход	Байт данных для вывода на печать
10 11	- BUSYP	- Вход	Не используется Высокий уровень сигнала указывает, что ПУ не может принять данные
12 13	- ACK	- Выход	Не используется Сигнал выбора печатающего устройства
14 21 22 23 25	- - Земля - -	- - - - -	Не используется Не используется Не используется Не используется Не используется

Сигналы на контактах розетки разъема адаптера последовательного интерфейса

Номер контакта	Наименование сигнала	Направление	Тип интерфейса	Описание сигнала
01	RCVD+	Вход	ИРПС	Принимаемые данные (ИРПС)
02	RXD	Вход	RS232	Принимаемые данные (сигнал идентичен цепи 103 ГОСТ 18145-81)
03	TXD	Выход	RS232	Передаваемые данные (сигнал идентичен цепи 104 ГОСТ 18145-81)
04	RTS	Выход	RS232	Сигнал готовности персонального компьютера (идентичен цепи 107 ГОСТ 18145-81)
05 06	Земля DSR	- Вход	- RS232	- Сигнал готовности внешнего устройства (идентичен цепи 108 ГОСТ 18145-81)
07	TRND+	Выход	ИРПС	Передаваемые данные (ИРПС)
08	TRND-	Выход	ИРПС	Передаваемые данные (ИРПС)
09	RCVD-	Вход	ИРПС	Принимаемые данные (ИРПС)

Сигналы на контактах розетки кассетного накопителя

Номер контакта	Наименование сигнала	Направление	Описание сигнала
01	STOP2	Выход	Цепь реле "кратковременный стоп"
02 03	Земля STOP1	- Выход	- Цепь реле "кратковременный стоп"
04	CASIN	Вход	Входной сигнал от кассетного накопителя
05	CASOUT	Выход	Выходной сигнал на кассетный накопитель

фейс типа RS-232. При работе с интерфейсом ИРПС необходимо установить переключку между контактами 04 и 02 соединителя адаптера. BIOS поддерживает только работу RS-232.

Адаптер печатающего устройства (имеется только в учительской ПЭВМ) осуществляет вывод данных на печать и реа-

лизует необходимое подмножество цепей параллельного интерфейса типа ИРПС-М (CENTRONIX). Это подмножество включает восемь однонаправленных цепей данных с высоким активным уровнем напряжения и цепи сигналов стробирования SE, выбора ПУ ACK и сигнала BUSYP.

DRAW на PMU

Для расширения графических возможностей ПК «Корвет», стоящих на PMU, предлагаю программу, отчасти заменяющую функцию DRAW.

```
10 LINE (470,20)-(470,60),7
20 LINE (440,40)-(500,40),7
30 LINE (440,20)-(500,60),7
40 LINE (500,20)-(440,60),7
50 CLS: PRINT "ВВЕДИТЕ
      КООРДИНАТЫ НАЧАЛЬНОЙ ТОЧКИ"
60 INPUT X,Y: PSET (X,Y),7
70 LOCATE 54,1: PRINT "H   U   E"
80 LOCATE 54,3: PRINT "L   R"
90 LOCATE 54,5: PRINT "G   D   F"
100 INPUT Z$: INPUT X1
110 IF Z$="L" THEN A=0-X1: B=0
120 IF Z$="R" THEN A=X1: B=0
130 IF Z$="U" THEN A=0: B=0-X1
```

```
140 IF Z$="D" THEN A=0: B=X1
150 IF Z$="F" THEN
      B=SQR(X1^2/2): A=B*1.5
160 IF Z$="E" THEN
      B1=SQR(X1^2/2): A=B1*1.5: B=0-B1
170 IF Z$="G" THEN
      B=SQR(X1^2/2): A=0-B*1.5
180 IF Z$="H" THEN
      B=0-SQR(X1^2/2): A=B*1.5
190 IF Z$="K" THEN 50
200 IF Z$="N" THEN 220
210 LINE -STEP(A,B),7: CLS: GOTO 70
220 BEEP
```

71

С. ГОЛОВКОВ

От редакции. Есть и другой путь решения этой проблемы, например пользоваться операционной системой «Корнет», позволяющей работать на PMU с Бейсиком РМП.

Копите деньги

В фирме ПараГраф (Москва) создана программа, распознающая рукописный текст. Она справляется даже с текстом, написанным на бумаге и затем введенным в компьютер; тем легче ей будет распознать буквы, вводимые непосредственно в момент написания на специальном чувствительном «блокноте» — ведь в этом случае к информации о форме добавляется информация о последовательности написания.

Алгоритм покупает фирма Apple, озабоченная созданием компьютера с бесклавиатурным вводом. Их техника плюс наш алгоритм — успех обеспечен! Так что если вам лень учиться печатать...



Берегитесь, самоделкищи!

Широкое использование самодельных модемов и автоматических определителей номера звонящего вам абонента повышает нагрузку на телефонные сети, а порой и выводит их из строя. Некоторые модели модемов создают помехи для соседних каналов связи. Разгневанные телефонисты намерены ужесточить контроль за подключением к сетям «нештатных» устройств.

Еще один «вставной» компьютер

Открытость архитектуры IBM PC позволяет множеству разработчиков предлагать все новые и новые платы, предназначенные для включения в базовую схему компьютера и увеличивающие его возможности. И если программное сжатие данных для увеличения эффективной емкости магнитных носителей требует времени, то сжатие аппаратное, с помощью сопроцессора Stacker фирмы Stack Electronics, выполняется в реальном времени, иначе говоря, незаметно для пользователя.

Плата Stacker, имеющая собственный микропроцессор, подключается к одному из резервных разъемов IBM PC и сжимает файлы в среднем в 2 раза.

Текстовый редактор для БК: новые принципы и возможности

Родоначальником текстовых экранных редакторов для БК-0010 был EDASP (остающийся и сейчас лучшим «карманным» редактором). Все остальные разработки дополняли или изменяли те или иные функциональные возможности EDASP: появились новые драйверы принтеров, улучшался сервис, варьировались системы директив. Изредка появлялись и оригинальные программы со своими оригинальными форматами, например редакторы микроассемблеров С. Кумандина и А. Сомова, обладающие рядом неоспоримых достоинств по сравнению с EDASP: малым объемом самого редактора, компактностью выходного файла, достигнутой за счет кодировки пробелов, хорошо продуманным набором команд и динамическим перераспределением памяти.

У всех существующих текстовых редакторов имеется общий недостаток — жесткая структура, не позволяющая пользователю перенастраивать программу под свои потребности и обрекающая его на необходимость иметь дело с несколькими редакторами. Это и послужило причиной создания редактора ЕДИТ2, в основу которого положен принцип модульности, аналогичный применяемому практически в любой современной операционной системе.

Пользователь ЕДИТ2 обладает базовым вариантом редактора с набором стандартных драйверов и имеет выбор дополнительных драйверов, с помощью которых может производить перенастройку (например, под свой принтер или экранный шрифт любого национального алфавита или начертания).

Драйверная система построена так, что пользователю не нужно беспокоиться о длине и адресе загрузки драйвера: программа сама его загрузит и установит в себе и все дальнейшие операции произведет под своим управлением.

При разработке ЕДИТ2 были учтены практически все достижения в области сервисных услуг и команд «карманных» текстовых редакторов. В первую очередь ликвидирован наиболее существенный недостаток

многих таких программ — отсутствие автоповтора при вводе символов с клавиатуры: в ЕДИТ2 все клавиши способны работать в режиме автоповтора (включается при удержании клавиши в нажатом состоянии более 1 с). Для купирования возможного «дребезга» клавиатуры введена процедура настройки драйвера клавиатуры во времени восприимчивости. В течение «периода невосприимчивости» драйвер нечувствителен к повторяющимся кодам, но воспринимает новые, что позволяет избавиться от «дребезга» без снижения скорости набора текста. Индикаторы свободной памяти обычно делаются в виде 4-значного числа, рассматривание которого, по замыслу, способно защитить пользователя от неожиданной нехватки ОЗУ. Между тем давно замечено, что подсознание человека, плохо реагируя на цифровую информацию, отлично воспринимает образную; пример — часы цифровые и обычные. Поэтому гораздо более удобен аналоговый линейный индикатор памяти, одного взгляда на который достаточно, чтобы определить, сколько текста еще можно вводить, не опасаясь за последствия. Именно такой индикатор реализован в ЕДИТ2 (первые на БК-0010).

Большие неудобства вызывает несовместимость форматов текстов, написанных с помощью разных редакторов. В ЕДИТ2 этой проблемы нет: перед вводом файла с магнитофона программа может быть настроена на переформатировку введенного текста в свой собственный формат.

При наборе таблиц и некоторых других текстов в каждой строке содержится много пробелов, «съедающих» память и сокращающих конечный размер документа. В ЕДИТ2 применен экономичный вариант кодировки пробелов символами ТАБ.

Обычно в редакторах — и EDASP не является исключением — ширина текста задается пользователем в виде числового параметра. Это мешает наглядно представить, как будет выглядеть текст на экране или при печати. В ЕДИТ2 задание длины строки осуществляется с помощью специаль-

ной мерительной линейки, по которой перемещается курсор, определяя правую границу строки. Одной командой при этом можно соотнести длину строки с набранным текстом и при необходимости установить новую длину строки.

При наборе текстовая строка в ЕДИТ2 постоянно находится в режиме автосвертки: слово, не помещающееся в установленных границах строки, автоматически переносится на следующую строку. За четыре знакоместа с правой границы строки подается звуковой сигнал.

Установленная длина строки используется и при форматировании текста. Процедуры форматирования две: «глобальная» (для всего текста) и выборочная (только для одной строки). Обе построены на одном принципе: выравнивание текста по правому краю путем добавления пробелов между словами до достижения установленной границы.

Выборочное форматирование удобнее при работе со строками, меньшими половины установленной длины строки. При необходимости эта процедура может стать и «глобальной», если использовать ее в рамках макрооперации с соответствующим числом повторов.

Команда записи текста на магнитофон позволяет (в отличие от EDASP) записать указанное число копий.

Нововведением является и наличие мощной системы транслитерации, которая в сочетании с макрокомандным режимом выводит ЕДИТ2 на качественно новый уровень сервисных возможностей.

Таблица кодировки символов БК содержит 4 набора латинских и русских букв, так что 12 видов транслитераций (большие латинские в малые латинские, большие латинские в малые русские и т. д.) на первый взгляд исчерпывают возможные варианты. Однако есть и тринадцатый вид — он предназначен для перекодировки маленьких букв в большие и обратно с учетом структуры текста: переводу не подлежат первые буквы предложений.

При возникновении нехватки памяти ЕДИТ2 автоматически переходит в режим расширенной памяти; при освобождении буфера текста происходит обратный переход. Таким образом, редактор позволяет создавать тексты длиной до 23 Кбайт.

В связи с тем что большинство пользователей привыкли работать с EDASP, в ЕДИТ2 сохранены основные управляющие клавиши и команды этого заслуженного редактора.

Основные параметры ЕДИТ2:
размер редактора — 3,5 Кбайт;

размер строчного буфера — 128 байтов;
размер буфера сохранения — 128 байтов;
размер буфера макро — 64 команды;
размер буфера модели — 64 байта;
размер буфера фрагмента — 1700 байтов;
размер буфера текста — 23 Кбайт.

Работа с программой. Режим 32 или 64 символа в строке устанавливается до запуска. После старта программа спрашивает «Драйверы стандартные?», на что следует ответить Д, если не будет загружаться внешние драйверы, или Н, если редактор будет перенастраиваться под нестандартные устройства — дисковод, другой принтер, новый знакогенератор экранного шрифта и т. д. В этом случае запрашивается имя драйвера, и после его загрузки он становится частью системы и работает под ее управлением. В настоящее время с редактором поставляются драйвер экранного шрифта IBM, драйверы принтеров MC 6312, MC 6313, D 100, драйвер печати графической копии 73 экрана и др.

После настройки запрашиваются имя файла для считывания и его формат.

Программа имеет две точки входа: холодный старт — адрес 1000; теплый старт — 1002. При запуске с адреса 1000 происходит начальная установка системных ячеек редактора и очищается буфер текста. Запуск с адреса 1002 позволяет перезапустить редактор с сохранением ранее набранного текста.

Процесс набора команд безразличен к состоянию регистров клавиатуры; положительным ответом на запросы служит нажатие клавиши ВК или любой последовательности клавиш, а отказом — любое слово, начинающееся с N.

Перечисление команд редактора начнем с обычных перемещений курсора на символ и строку, осуществляемых клавишами-стрелками; если предварительно нажать клавишу КТ, курсор перемещается в начало (конец) строки или вверх-вниз на экранную страницу. Имеется также возможность перемещения курсора в позицию текста, кратную 8 (с помощью клавиши ТАБ), перехода в начало предыдущего или конец следующего слова, в начало следующей строки, начало текста, начало последней страницы текста.

Следующая группа команд — стирание, восстановление, вставка: стирание символа перед курсором с подтягиванием остатка строки, восстановление удаленной строки или ее части, стирание символа в позиции курсора с подтягиванием остатка строки или сцеплением со следующей строкой, стирание строки слева (справа) от курсора с занесением фрагмента в буфер, перемеще-

ние остатка строки вправо на один символ (вставление пробела), разбиение строки текста на две, вставка пустой строки, удаление и вставка фрагмента текста.

Клавишами **KT+P** устанавливается режим вставки-замены. Режим замены индицируется в служебной строке.

Можно задать автоматическое выполнение (определенное число раз) какой-либо команды, задать макропоследовательность команд и исполнить ее. Имеются команды ввода модели и ее поиска в тексте, установки метки и перехода на ранее установленную метку, выравнивания строки или фрагмента текста по правому краю в соответствии с установленной длиной строки. Имеется также возможность задать размер (число строк) экранной страницы. По завершении работы текст можно записать на магнитофон или распечатать.

Дозагружаемые модули можно использовать готовые, а можно и написать самостоятельно. В связи с этим приведем некоторые сведения, полезные при написании собственных драйверов для данного редактора.

Адреса с 400 по 500 (в восьмеричной системе счисления) используются основным модулем, и всякое вмешательство в эту область может привести к нежелательным последствиям.

Адрес	Назначение
400	Адрес-указатель на начало текущей строки, т. е. строки, в которой находится курсор
402	Адрес начала буфера текста; эта величина непостоянна и зависит от длины драйвера, загруженного вместе с редактором
404	Адрес конца текста, зависит от длины введенного текста
454	Длина страницы для печати (после вывода указанного числа строк программой выдается код перевода формата)
456	Величина отступа от левого края при печати
462	Используется драйвером загружаемого знакогенератора как указатель на начало таблицы образов символов
464	Используется драйвером принтера. Нулевое содержимое указывает на применение встроенного драйвера печати, ненулевое — на адрес начала загруженного драйвера печати
466	Длина загруженного драйвера

На загружаемый драйвер накладываются следующие ограничения:

1. Драйвер должен быть написан в перемещаемом формате, т. е. работать в любой области памяти.

2. Тело драйвера должно заканчиваться командой **RTS PC**.

3. Структура драйвера должна соответствовать приведенной ниже.

СЛОВО-ИДЕНТИФИКАТОР МОДУЛЬ ИНИЦИАЛИЗАЦИИ ТЕЛО ДРАЙВЕРА

Здесь слово-идентификатор — слово с числом 252 в младшем байте, старший байт зарезервирован; модуль инициализации — модуль, устанавливающий драйвер в системе и передающий ему управление; тело драйвера — сама программа.

В качестве примера приводится текст эмулятора знакогенератора экранного шрифта IBM.

```

;-----
BUFUS: .WORD 252 ; БАЙТ-ИДЕНТИФИКАТОР
;-----
; МОДУЛЬ ИНИЦИАЛИЗАЦИИ
MOV PC,R5
ADD (PC)+,R5 ; ПЕРЕХВАТ
; ЗАПРОСОВ EMT

.WORD @EMT@+2
MOV R5,@#3@
ADD #DRITT-EMT@,R5 ; ПОЛУЧЕНИЕ
MOV R5,@#462 ; АДРЕСА
; ТАБЛИЦЫ
; ОБРАЗОВ
; СИМВОЛОВ

RETURN

;-----
; ТЕЛО ДРАЙВЕРА
EMT@: MOV R5,-(SP)
MOV 2(SP),R5
MOV -(R5),R5
BIC #1774@,R5
CMP R5,#16 ; ПЕРЕХВАТИТЬ
BNE 3@ ; ЗАПРОС EMT 16
JSR R4,@#11@346 ; И ОБРАБОТАТЬ
; ЕГО СОГЛАСНО
; НОВОЙ ТАБЛИЦЕ
; ОБРАЗОВ

BIC #1774@,R@
CMPB R@,#4@
BSC 2@
CMPB R@,#2@
BSC 1@
CMPB R@,#24@
BSC 2@
SUB #4@,R@
SUB #4@,R@
1@: ASL R@
MOV R@,R1
ASL R@
ADD R@,R1
ADD @#462,R1 ; ПОЛУЧЕНИЕ
; АДРЕСА
; СИМВОЛА
CALL @#1@2764 ; ВЫВОД СИМВОЛА
CALL @#1@2374 ; НА ТЕРМИНАЛ
; В РЕЖИМЕ
; ТЕЛЕТАЙПА

JSR R4,@#11@362
MOV (SP)+,R5
RTI
2@: JSR R4,@#11@362
3@: JMP @#1@@12@

```


BIGRED — текстовый редактор для БК-0010(01)

BIGRED является дальнейшим развитием текстовых редакторов EDASP для компьютеров БК-0010 и БК-0010-01, позволяющим более чем вдвое превышать размеры стандартного текстового файла для БК (как по длине строк, так и по объему обрабатываемого текста) и эффективно управлять принтером при печати.

EDASP (последняя известная нам версия выпущена в феврале 1988 г.) при малых размерах имеет богатый выбор возможностей для обработки текста, надежен в работе, прост для изучения и стал фактически стандартом текстового редактора для БК. К сожалению, группа ASP перестала поддерживать его, а доработки других программистов страдают локальностью и улучшение в них какой-то одной функции часто приводит к внесению ошибок в другие.

BIGRED способен делать все то, что делает обычный EDASP: работать в режимах ВСТАВКА или ЗАМЕНА; перемещать курсор на символ, слово или строку, в начало или конец строки, на заранее отмеченную строку, в начало или конец текста; сдвигать и раздвигать символы в строке; забивать символы; стирать строку или ее части с возможностью восстановления стертого; преобразовывать строчные буквы в прописные и наоборот; осуществлять горизонтальную таблицу; удалять части текста с возможностью сохранения его в буфере или без сохранения; при превышении строкой заданной длины автоматически переносить на новую строку части слов; повторять группы команд заданное число раз; вести поиск группы символов в тексте по заданной модели; задавать макрокоманды (цепочки команд); работать с магнитофоном и принтером.

Действие команд может отличаться от «стандартного» только за счет упрощения работы с ними, устранения ошибок или расширения возможностей, о чем сказано ниже, а назначение клавиш взято из версии EDASP10 (20.02.88, А. В. Шакунов.).

Основным назначением стандартного EDASP было создание текстового файла

на магнитной ленте. Однако неудобство работы с магнитофоном привело к попыткам объединить все операции обработки текста в одной программе.

Возможности распечатки текста прямо из редактора наиболее велики у программ EDASPS (К. В. Косов) и EDA (А. С. Лозовицкий). Эти программы и были взяты за основу при разработке алгоритма работы редактора BIGRED с принтером.

75

1. Печать на принтере ведется с отступом от левого края листа. Величину этого отступа можно изменять построчно в широких пределах.

2. Управление режимами работы принтера осуществляется вводом в процессе подготовки текста специальных управляющих кодов.

3. В зависимости от типа принтера применяются различные версии BIGRED, позволяющие наиболее полно использовать возможности данного принтера.

Так, при работе с принтером Robotron CM 6329 (и совместимых с ним по кодам: D-100M, «Электроника MC 6312» и др.) реализована оригинальная разработка, позволяющая выводить на печать любой символ из набора символов БК, включая все символы псевдографики. Размеры символов согласованы с размерами основных шрифтов принтера; различие в начертании букв принтера и БК создает дополнительные полиграфические возможности при печати.

BIGRED способен загрузить с магнитофона для дальнейшей распечатки или редактирования практически любой файл. Ограничением является длина файла: если она превышает 50 000⁸, теряется часть последних символов. Загружаться может не только текстовый файл, написанный в формате EDASP, но любой, содержащий текстовые вкрапления. При этом недопустимые в формате EDASP символы автоматически заменяются на пробелы, и пользователю остается только стереть «мусор», появившийся на месте нетекстовой части файла.

При работе в режимах ЧТЕНИЕ и ЗАПИСЬ предусмотрено повторение опера-

ций без повторного набора имени: это удобно при некачественном считывании файла с МЛ и при повторной записи дубля файла на МЛ.

BIGRED имеет встроенный каталогизатор, что может быть оценено теми, кто следит за своей библиотекой программ: это позволяет устранить лишнюю операцию записи на магнитофон для редактирования или распечатки каталога.

Чем еще отличается BIGRED от EDASP? Помимо преобразования заглавных букв в строчные и наоборот в BIGRED добавлена оригинальная операция преобразования русских букв в латинские и наоборот, зачастую необходимая при редактировании; длительное нажатие любой клавиши вызывает автоматическое повторение ее действия; реализован оригинальный алгоритм автоматического перехода редактора в режим расширенной памяти и обратно в зависимости от объема обрабатываемого текста. Использование режима расширенной памяти позволяет обрабатывать BIGRED тексты, длина которых вдвое превышает возможности EDASP (более 21 500 символов).

Максимальная длина строки в BIGRED вдвое превышает длину строки экрана. Это позволяет эффективно использовать принтер

с широкой кареткой. О выходе правого края строки за пределы экрана свидетельствует стрелка-признак на последнем знакоместе текущей строки. Стрелка не влияет на набранный текст. Попутно с введением в редактор этого оригинального признака исправлена ошибка EDASP, из-за которой в нем иногда изменяется порядок символов в выходящей за край экрана части строки.

В режиме команды ПОВТОР в служебной строке редактора показывается убывающий десятичный счетчик, облегчающий контроль за работой при большом числе повторений долго выполняющихся команд.

Переключение режимов ВСТАВКА и ЗАМЕНА сделано проще и с постоянной индикацией текущего режима.

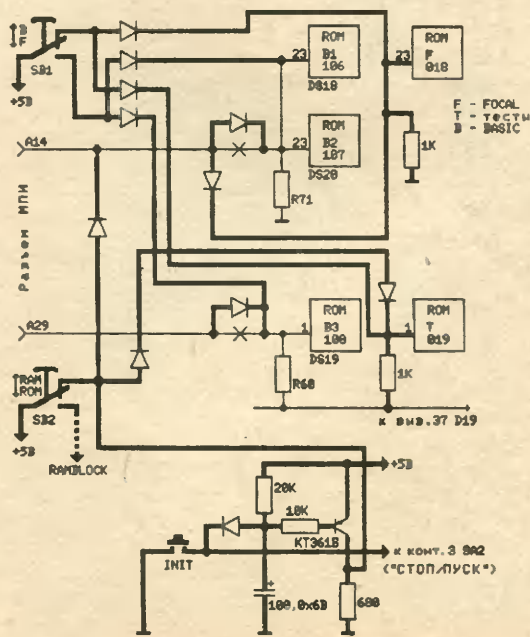
При сравнении текстовых редакторов основными являются два параметра: объем обрабатываемого текста и удобство работы. Если оценивать по ним BIGRED, то следует отметить, что размер текста, обрабатываемого им до перехода в режим расширенной памяти, не меньше, чем в стандартном EDASP, а максимальный размер обрабатываемого текста вообще не имеет аналогов для БК.

А. БАКУЛИН, В. ЧЕРНЯКОВ

Попробуйте так!

Ни для кого не секрет, что БК-0010(-01) — это конструктор для умелых рук. В «ИНФО» публиковалось немало рекомендаций по его доделкам и усовершенствованиям.

Главное место в этом длиннейшем списке, бесспорно, занимает способ избавления от блока МСТД. В [1] описано, как это сделать. Существует и менее кропотливый способ: микросхемы (МС) ПЗУ с Фокалом 018 и тестовой диагностикой 019 выпаивают из платы МСТД и напаивают «этажеркой» на МС ПЗУ Бейсика с номерами 106 и 108 соответственно. При этом выводы 23 МС 018 и 1 МС 019 отгибают и распаивают гибким монтажным проводом согласно схеме, приведенной на рисунке. Необходимые коммутации осуществляются двумя переключателями: с помощью SB1 выбирают ПЗУ Бейсика или Фокала и тестовой диагностики, а SB2 коммутирует дополнительные ПЗУ или блок расширения ОЗУ. При их отсутствии этот переключатель спасает от неприятных сюрпризов при работе в кодах или на ассемблере, порождаемых неприятной «привычкой»



БК к перезапуску системы, приводящему к искажению информации в ОЗУ. В предлагаемой схеме также решена проблема зависания (зацикливания) БК: достаточно нажать кнопку «INIT», и произойдет перезапуск процессора с адреса «холодного» старта БК (100 000), а каскад на транзисторе VT1 обеспечит блокировку ПЗУ на время перезапуска. В результате управление останется в мониторе, а содержимое рабочей области ОЗУ не пострадает. Директивы «MONIT» Бейсика и «Р М» Фокала становятся просто ненужными. Кроме того, выходить в монитор после включения БК после такой переделки уже нет необходимости.

К сожалению, попытка повторно войти в Бейсик по адресу «теплого» старта (120300) после снятия зависания удастся не всегда.

Полное наименование MC ПЗУ с Фокалом — K1801PE1A-018. Здесь и далее они обозначаются только по трем последним цифрам, определяющим записанную в них информацию. В некоторых блоках МСТД вместо MC 018 может быть установлена MC 084. Все позиционные обозначения элементов БК даны по его схеме, приведенной в [1]. Следует учесть, что в БК-0010-01 подключение ПЗУ несколько отличается, а именно: на месте панельки XT8 установлена MC ПЗУ с позиционным обозначением DS20;

типы ПЗУ DS18—DS20 соответственно K1801PE2A-106, 107, 108; выводы 23 DS18 и DS20 подключены к общему проводу через резистор R71 (1 кОм) и выведены на контакт A14 разъема МПИ; вывод 1 DS19 подключен к контакту A29 разъема МПИ.

Все эти отличия учтены в приводимой схеме.

Для монтажа лучше всего подходит резервное место под микросхему в нижнем левом углу платы БК: на его диагональ выведены общий провод и питание +5 В. Сигнал, блокирующий ПЗУ (ROMBLOCK), подается на верхний (по схеме) вывод SB2. Если блок МСТД хотят оставить в

неприкосновенности, то сигнал ROMBLOCK следует подать через два диода на выводы 23 и 1 MC DS18 и DS19 аналогично тому, как это показано на схеме.

Если не было ошибок при монтаже, то никакой настройки не потребуется, кроме, быть может, увеличения емкости C1 при нечеткой блокировке ПЗУ. Все диоды типа КД521 или КД522, транзистор может быть заменен на любой *p-n-p* типа с коэффициентом усиления не ниже 50, номиналы остальных элементов могут отличаться от указанных на 50 %.

Несколько слов о реализации turbo-режима [2]. Прежде чем начинать такую переделку, нужно убедиться в ее возможности для данного экземпляра БК. Для этого аккуратно перережьте дорожку, идущую от D8.2 (вывод 9) на вход CLC процессора (вывод 1), и соедините перемычкой вход CLC с выводом 5 или 13 D8. Включите питание БК и проверьте его работоспособность: если происходит зависание, попробуйте уменьшить (вплоть до нуля) сопротивление R28, а если происходят сбои при записи в системный регистр (самопроизвольно срабатывает реле управления магнитофоном, искажается или полностью исчезает звук), попробуйте увеличить емкость C8 до 300—500 пФ. После этого закройте корпус и проверьте работу БК в течение часа-полтора. Если результаты испытаний будут удовлетворительными, можно начинать переделку всерьез.

Литература

1. Панченков И. Приручение БК // Информатика и образование. 1990. № 4. С. 68.
2. Антонов Д. Увеличение тактовой частоты // Информатика и образование. 1990. № 2. С. 46.

С. КАРАГИОЗ

От редакции. Предлагаемый способ, в отличие от описанного в [1], можно рекомендовать только тем, кто в совершенстве владеет паяльником и хорошо знаком с правилами монтажа интегральных схем. Неквалифицированная попытка реализовать этот способ может нанести компьютеру непоправимый вред.

Подключение принтеров к БК-0010(01)

Несмотря на большое число публикаций о подключении принтеров к БК, многие читатели испытывают затруднения. Чтобы помочь им, мы публикуем систематическое изложение этого вопроса.

Появившиеся наконец-то в свободной продаже относительно дешевые принтеры, большое число имеющихся драйверов для печатающих устройств и богатые возможности БК по подключению разнообразной периферии позволяют перевести использование этих компьютеров на качественно новый уровень. БК с принтером — серьезный рабочий инструмент творческой личности. Показателен в этом смысле пример известного американского писателя-фантаста А. Азимова, который никогда не страдал от низкой производительности своего труда, однако, научившись работать на текстовом процессоре, горестно заявил, что вся его прошлая литературная жизнь — черепашья прогулка по полю незатронутых тем, а настоящая работа только теперь и начинается...

Для начала разберемся, как соединить печатающее устройство (ПУ) с БК, — редакционная почта показывает, что большинство вопросов возникают на этом этапе.

ПУ подключаются к компьютерам с помощью одного из двух типов каналов связи: параллельного (ИРПР, ИРПР-М, IFSP) или последовательного (ИРПС, IRPS) с протоколом обмена RS-232 или CENTRONIX. У наиболее совершенных ПУ и компьютеров имеются оба типа каналов; именно к этому классу относится БК. Однако исторически сложилась практика использования у БК почти исключительно параллельного канала. На сегодня известно о подключении по последовательному каналу только принтера «Robotron» CM 6329.01 (П. Чирков, г. Львов).

Для начала кратко опишем параллельный и последовательный интерфейсы. Основой параллельного в БК является программируемый порт ввода-вывода (разъем XT5, УП), доступ к которому осуществляется с помощью двух разных (несмотря на одинаковый адрес — 177714) регистров — чтения и записи. Соответствие контактов разъема XT5 сигналам порта приведено в таблице 1. Программной поддержки в штатном мониторе порт не имеет, и в большинстве случаев приходится использовать загружаемый драйвер принтера.

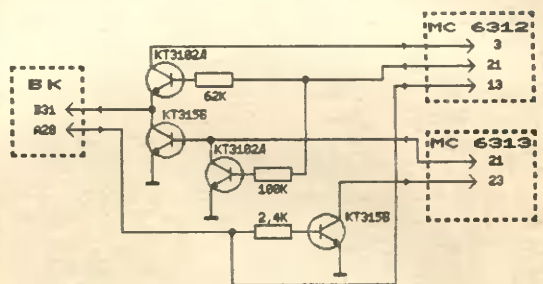
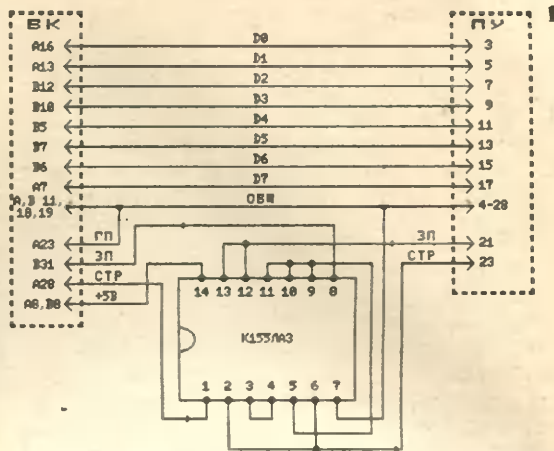
Общая схема работы драйверов такова. Проверяется состояние 8-го разряда регистра чтения порта ввода-вывода (сигнал BUSY). Если он равен 1 (этому соответствует нулевой потенциал на контакте В31 разъема XT5), то это означает, что ПУ готово принять очередную байт данных. В таком случае драйвер записывает в младший байт регистра записи порта очередной байт данных, а в 8-й разряд регистра записи (сигнал STROBE) — 1. Последнее является сигналом разрешения считывания данных для ПУ. ПУ считывает данные и заносит 0 в ЗАН, а драйвер после этого заносит 0 в СТРОБ. Так работает, в частности, драйвер из ПЗУ с Бейсиком.

Для наиболее распространенных ПУ (соответствия их контактов контактам XT5 приведены в таблице) существует уже множество драйверов — оформленных в виде отдельных программ или входящих в состав редакторов (текстовых и графических) и каталогизаторов-верификаторов. Перечислим некоторые из них:

D100 — PRINTER, PRINTFOC,
 PRINTER2, EDASP10;
 УВВП4 30-004 — BED, EDASP10M9,
 GRATEP, BIGRED, EDIT.WS;
 ROBOTRON — ROB.IRPS, ROB.IFSP,
 BIGRED, GRAFRE, TED6+ROB64.DRV,
 TED6+ROB32.DRV;
 MC 6312 — ПМ6312, ПС6312, MS6312.SAV,
 BIGRED, EDIT4.WS+LP12, ТСТФ6312,
 VERIFY+;
 MC 6313 — ПМ6313, ПС6313, MS6313.SAV,
 EDIT4.WS+LP13, ТСТФ6313, VERIFY+,
 P1.0M+ГПКМ6313;
 Think Jet — SCREEN.BIN;
 GEMINI-10XR — ПМ6312, ПС6312;
 EPSON — TED6+EPSON8.DRV, TED6+
 +EPSON7.DRV.

Заканчивая тему параллельного интерфейса, приведем две полезные схемы. Первая (рис. 1) — инвертирование сигналов STROBE и BUSY, необходимого для некоторых комбинаций принтер — драйвер (например, для «Электроники MC6313» и драйвера PRINT2). Вторая (рис. 2) — оригинальное

Контакты разъема "УП"	Обозначение сигнала	Наименование сигнала		Robotron СМ 6329.02	ЕС 7189 СМ 6325 D-100	УВВТЧ 30-004	З л о т р о м и к а		Think Jet	Gemini 10XR
		русское	международное				MC 6312	MC 6313		
A16	В000	D0	DATA1, D1	B5	2	B14	2	3	1	2
B12	В002	A2	DATA3, D3	B6	3	B13	3	5	2	3
B10	В003	A3	DATA4, D4	B7	4	B12	4	7	3	4
B5	В004	A4	DATA5, D5	B8	5	B11	5	9	4	5
B7	В005	A5	DATA6, D6	B9	6	B10	6	11	13	6
B6	В006	A6	DATA7, D7	B10	7	B8	7	13	14	7
A7	В007	A7	DATA8, D8	B11	8	B9	8	15	15	8
A28	В008	СТР	STROBE, SC, DAV	B12	9	B7	9	17	16	9
B31	В008	ЭП, ЗАН	BUSY, AC, NDAC	B2	10	B6	10	23	6	11
A, B 11, 18, 19	ОБЩ	ОБЩ	GND, 50, Logicbnd	B4, A4	19, 21	B1, B16, A1	15	4, 6, 8, 10, 12, 14, 16, 18, 20	24	16, 19-30

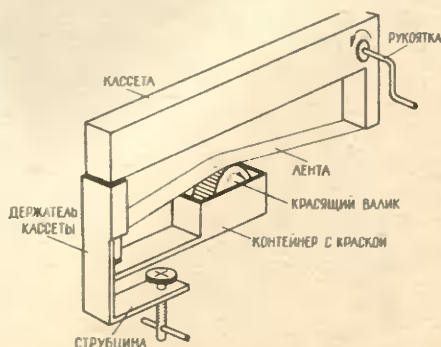


объединить достоинства струйного и матричного принтеров.

О последовательном интерфейсе достаточно хорошо рассказано в журнале «Вычислительная техника и ее применения» № 7 за 1991 г.

А теперь немного об «узких» местах принтеров. Первое из них — лента: она либо рвется, либо высыхает.

Что делать, если лента порвалась? Советую в предвидении такой неприятности сделать себе очень простое и удобное приспособление для ее сварки, показанное на рис. 3. Способ применения ясен из рисунка и в дополнительных пояснениях не нуждается.



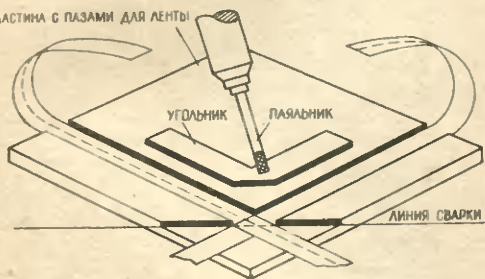
подключение (разработанное С. Карагиозом, Москва) к порту ввода-вывода сразу двух ПУ — если вам, например, захотелось

Сделать же устройство можно из чего угодно: картона, текстолита, оргстекла — важно только, чтобы под жалом паяльника была полоска металла (жести, фольги).

Ну а что делать, если лента высохла? Восстановить ее свойства очень просто — достаточно равномерно пропитать ее каким-нибудь нейтральным маслом: вазелиновым, трансформаторным или любым другим, не дающим кислой реакции, добавив к нему 10 % глицерина. Проще всего пропитку сделать при помощи куска поролона, сквозь который следует протащить всю ленту несколько раз, пока она не пропитается равномерно и негусто маслом. Ю. Зальцман рекомендует всунуть при этом руки в полиэтиленовые пакеты, чтобы не испачкаться краской. При неинтенсивной работе принтера этот способ, конечно, хорош своей простотой и «незатратным» механизмом. Однако, на мой взгляд, лучше не пожалеть времени и постараться сделать специальный станок для пропитки лент не только маслом для освежения, но и краской для продления сроков службы ленты в 50—100 раз (!). Принцип действия такого станка чрезвычайно прост и ясен из рис. 4. Единственное, что нужно для его безупречной работы, — достаточно жидкая консистенция раствора типографской краски (в том же масле с примесью глицерина): она должна приближаться к консистенции магазинного молока. Достоинством такого подхода является быстрота операции и чистота рук.

80

4 ПЛАСТИНА С ПАЗАМИ ДЛЯ ЛЕНТЫ



У струйных принтеров (МС 6312 и т. п.) узким местом является емкость чернильницы, которой хватает на 300—350 страниц текста. Как заправить чернильницу? Опять

же весьма просто: нужно взять обычные чернила «Радуга», добавить к ним 30 % по объему этилового спирта (водки) и 5 % (тоже по объему!) глицерина, прокипятить эту смесь 10—15 мин на слабом огне и профильтровать через слой ваты. Заправку головки лучше всего вести следующим образом: держа головку матрицей вверх, осторожно чем-нибудь тупым сжать пластиковый мешок чернильницы до появления чернил на матрице, а затем, капая пипеткой на матрицу приготовленные чернила, медленно отпустить мешок, чтобы он, расправляясь, засасывал чернила внутрь. Чернила при этом дополнительно фильтруются, головка не портится и может быть использована вплоть до физического износа термоэлементов матрицы (в среднем головка отечественного производства выдерживает 30—35 заправок, импортная — 70—80). Справедливости ради нужно упомянуть и другие способы заправки: один связан с проколом пластикового мешка шприцем с чернилами, а другой — со вскрытием заправочного отверстия на лицевой панели головки. Оба способа рано или поздно приводят к течи через сделанные отверстия и порче каретки. Пользоваться ими советую только очень нетерпеливым и... очень богатым.

Еще один совет дает И. Шторк (г. Алмата). Если головка не работала больше 3—4 ч, обязательно промойте ее перед работой, выдавив каплю чернил на матрицу и засосав ее обратно в мешок; остатки чернил сотрите ватой, смоченной спиртом. При заправке бумаги в принтер следите, чтобы головка находилась в крайнем левом положении: если этого не сделать, то есть опасность отрыва металлической пластины матрицы краем заправляемого листа. После заправки бумаги головку нужно осторожно вернуть в исходное положение, иначе сдвинется поле печати.

И в заключение совет самого общего характера: каким бы принтером вы ни располагали, помните — это сложный, высокоточный прибор, требующий квалифицированного и бережного обращения!

Авторами статьи «LinkWay — мультимедиа для IBM PC», опубликованной в ИНФО № 6—91, являются М. Маткин и Г. Полилова. Телефон для справок в Москве: 207-04-89.

Необходимость подключения к БК-0010 НГМД очевидна, и контроллеры, предназначенные для этого, производят многие кооперативы. Но, похоже, большинство производителей продают нечто, напоминающее грамофонные трубы без самого грамофона. Для эффективной работы с дисками требуется не только драйверок, позволяющий что-то куда-то записывать и читать, и не только мониторчик, который дает возможность читать файл в память и т. д., а операционная система, пусть с простейшими функциями, но имеющая тем не менее нормальные командный и программный интерфейсы, допускающая расширение и «прозрачная» для ранее написанных программ. Весьма желательна также совместимость хотя бы по форматам дисков и файловой системе с ближайшими родственниками БК — машинами ряда ДВК и УКНЦ.

Такая система была разработана автором статьи и в настоящее время выпускается вместе с контроллером гибких дисков малым предприятием «КомКон». К концу 1991 г. будет начат серийный выпуск на одном из заводов, производящих БК-0010.

Ниже приведено краткое описание ОС БК.

Схемы контроллеров для БК не отличаются большим разнообразием — в большинстве за основу взят аналогичный контроллер от УКНЦ. С минимальными изменениями эта же схема применена и в описываемой конструкции. К собственно контроллеру необходимо добавить ПЗУ и ОЗУ операционной системы, поместить все это в корпус — и можно подключаться к разъему системного канала БК-0010. Трудности и разноречивости в подходах начинаются далее.

Драйвер диска, форматировщик. Автору статьи совершенно непонятны мотивы, побуждающие многих программистов изобретать собственные форматы, лишаящие их разработки совместимости. Для БК допустим только один формат — тот, который используется в ДВК и УКНЦ (дискеты этого формата можно также читать специальными программами и на IBM PC). Напомним его: запись ведется методом модифицированной фазовой модуляции, что позволяет разместить на 80 дорожках двухстороннего диска 800 Кбайт информации; каждая дорожка разбита на 10 секторов по 512 байтов.

Драйвер обеспечивает обмен информацией между ОЗУ и диском блоками по 512 байтов, всего до 64 Кбайт за одну операцию,

и скрывает таким образом секторно-дорожечную структуру диска.

С драйвером тесно связан форматировщик, который предназначен для начальной разметки диска.

Файловая система. Блок является слишком низкоуровневым понятием, для пользователя ОС общепринятой единицей обмена информацией является файл. Под файлом мы понимаем непрерывный массив блоков на диске, как и в ОС ДВК. Для поддержки файловой системы каждый диск содержит свой каталог — список имен файлов и их «адресов». ОС имеет программный интерфейс для работы с файлами.

Программный интерфейс ОС БК. Программы имеют доступ к операционной системе через командные прерывания EMT. Набор прерываний, определенный в мониторе БК-0010, в ОС БК расширен и частично модифицирован. Например, EMT 36 в ОС БК обеспечивает операции не с магнитофоном, а с диском. Благодаря этому программы, разработанные в расчете на использование магнитофона, работают и в среде ОС БК, «не замечая» подмены его на диск. Для доступа к магнитофону введен запрос EMT 37, имеющий тот же формат.

В дополнительный набор прерываний входят запросы, предназначенные для доступа к файловой системе, клавиатуре, экрану, принтеру, а также для преобразований типа «число — строка» и RADIX — ASCII. Все запросы, аналогичные имеющимся в ОС ДВК, полностью соответствуют последним по форматам, что дает возможность разрабатывать программы для ОС БК в среде ДВК с использованием стандартных средств программирования — макроассемблера, системной макробιβотеки и т. д. Многие сервисные программы можно даже полностью отладить на ДВК.

Программист, пишущий на БК, может не знать о всех возможностях ОС и работать точно так же, как и без нее, но гораздо быстрее — благодаря диску. Кстати, сама ОС БК разрабатывалась на ДВК и активно использует командные запросы его ОС.

Командный интерфейс. В ОС БК принят простейший способ взаимодействия пользователя с системой, обычно называемый командным интерфейсом. Пользователь вводит с клавиатуры команды с необходимыми аргументами, система исполняет их. Программы, осуществляющие обработку команд, называются утилитами. Встроенные утилиты записаны в системном ПЗУ; поль-

зователь может добавить к ним внешние, располагающиеся на дисках, и завести таким образом свой собственный набор утилит.

В ПЗУ размещены лишь наиболее необходимые программы (просмотр каталога диска, копирование, удаление, переименование, печать файла, копирование с магнитофона на диск и наоборот и т. д.). Здесь же имеется редактор командной строки, позволяющий просматривать ранее введенные команды, выбирая нужную.

Ядро системы. ОС БК не нуждается в загрузке: она резидентна в адресном пространстве БК-0010 и автоматически запускается при включении питания компьютера, сразу же выдавая подсказку к вводу команды. В адресном пространстве оставлен свободным 6-й банк, на него система пытается передать управление после запуска. Это сдела-

но для того, чтобы, во-первых, можно было установить ПЗУ со своим пользовательским интерфейсом, а во-вторых, использовать контроллер с ОС БК в ранее разработанных на базе БК системах, для чего потребуется лишь перенести их встроенное ПО из 5-го в 6-й банк.

Описанная система вместе с контроллером представляет собой законченное устройство, превращающее БК-0010 в самый дешевый 16-разрядный компьютер, доступный сегодня потребителю. Она обладает неоспоримыми преимуществами по сравнению с аналогами и позволяет активно использовать как БК в качестве «домашнего ДВК», так и ДВК в качестве системы разработки программ для БК.

О. БУЛАНОВ

Программное исполнение системных команд Бейсика БК

Вильнюсская версия Бейсика, с которой приходится иметь дело пользователям БК-0010, не позволяет исполнять системные команды (SAVE, CSAVE, LIST и др.) программным образом. Это сильно обедняет возможности программистов и не позволяет автоматизировать некоторые весьма полезные процессы, например считывание или запись программ под контролем компьютера, просмотр текстов программ в автоматическом режиме и т. п.

Для ликвидации этого недостатка предлагается способ введения командной строки в текст программы и вызов ее на исполнение, иллюстрируемый демонстрационной программой.

```

10 CLS
20 PR="CSAVE"+CHR$(34)+"EXE"+CHR$(34)+
   CHR$(10)+"GOTO270"
30 ADR%=&01000
40 IF LEN(PR) MOD 2=1 THEN PR=PR+" "
50 FOR LX=1 TO LEN(PR) STEP 2
60 CD%=256*(ASC(MID$(PR,LX+1,1)))+
   (ASC(MID$(PR,LX,1)))
70 POKE ADR%,CD%
80 ADR%=ADR%+2%
90 NEXT LX
100 POKE ADR%,&012
110 POKE &0124,&01000
120 POKE &0122,LEN(PR)+1%
130 END
...
...
270 ?"Произведена запись
   в файл EXE.COD"
280 END

```

Комментарии к программе.

1. В строке 20 сформирована командная строка, содержащая команды CSAVE "EXE" и GOTO 270. После выполнения первой из них текст программы, содержащей эту строку, автоматически будет записан в файл с именем EXE в формате команды CSAVE. Затем последует уведомление оператора о выполненной процедуре.

В командной строке может быть записано столько команд, сколько позволяет буфер строки (255 символов). Нужно только учесть, что все команды в строке должны быть разделены кодом 10 (CHR\$(10)).

2. В строке 30 устанавливается адрес командной строки в памяти; стандартным является адрес 1000(8). Использовать другие адреса нецелесообразно, поскольку они либо используются программами, либо ненадежны, как, например, адрес буфера оператора INPUT 2422(8).

3. В строке 40 длина командной строки увеличивается до четного числа для удобства размещения в памяти.

4. В строках 50—90 размещается цикл, заносающий символы в память. Внутри цикла (строка 60) происходит превращение двух символов, идущих подряд, в одно число (форматирование записи).

5. В строках 100—120 осуществляется запись кода 10 в конце командной строки, установка адреса строки в памяти и ее длины.

6. Для вывода командной строки на экран

и ее исполнения обязательна строка 130 с командой END.

7. Остальные строки введены в демонстрационных целях для контроля выполнения командной строки.

Особенностью предлагаемого способа яв-

ляется то, что пока инициализированная строка команд не будет выполнена полностью, ее содержимое будет попадать в буферы операторов INPUT и INKEY.

А. МУМИНОВ

Кирпичики ваших программ

Рубрика с этим названием впервые появилась в нашем журнале в третьем номере за 1991 г. и немедленно вызвала большое число одобрительных отзывов у пользователей БК.

В этом номере, уважаемые читатели, вы найдете новую подборку программистских находок; надеюсь, что и на нее вы откликнитесь с не меньшим энтузиазмом.

И. Шебалкин предлагает вводить бегущую строку следующей подпрограммой:

```
1000 A$=STRING$(&H20,&H20)+A$+
      STRING$(&H14,&H20)
1010 I$LEN(B$)=1%THAX=1%ELAX=A$+1%
1020 B$=""+MID$(A$,AX,&H20)
1030 ?AT(0%,10%)B$
1040 IF INKEY<>"*THRETELGOT1010
```

Строка задается переменной А в любом месте основной программы, а темп ее вывода на экран может быть задан параметром N% в элементарном цикле задержки, встроенном в подпрограмму между строками 1020 и 1030:

```
1023 FOI%=0%TON%
1024 NEXI%
```

Этот же автор предлагает простой способ защиты программы от несанкционированного доступа: достаточно ввести в начало уже отлаженной программы строку

```
POKE 4%,&0134602
```

и любая попытка остановить работающую программу клавишей СТОП приведет к исполнению команды NEW со всеми вытекающими отсюда последствиями...

Д. Баженов предлагает свой прием ввода сообщения в служебную строку.

```
10 IFPEEK(&H20)=0%TH?CHR$(&H9B)
20 ?CHR$(&H94)CHR$(&H9E)CHR$(&H91)
30POKE&H82,&H3C00
40 ?AT(0%,0%) "... сообщение..."
50 POKE&H82,&H4000
```

Если к этой подпрограмме добавить три строки

```
60 POKE &H8B,&H4000
70 POKE &HFFB4,&H200
      OR (PEEK (&H84) /&H40-&H28)
80 LOC&NA,&NA
```

то служебная строка с сообщением сместится в низ экрана.

Предлагаемая им же подпрограмма позволяет вдвое увеличивать размеры символов на экране.

```
5 ?CHR$(140%)CHR$(140%)
10 DATA &H1345,&H15C3,&H4180,&H65C3,
      &H280,&H7F43,&H10C0,&H65C0,&H280,
      &H15C1,&H8A,&H15C2,&H20,&H140B,
      &H14F3,&H3E,&H7E84,&H65C3,&H40,
      &H7E49,&H87
20 FOAX=512%TO 552%ST2%
30 REAC%
40 POKEAX,C%
50 NEXAX
60 DEFUSR=512%
70 ?AT(3%,5%)"УДВОЕННЫЙ ШРИФТ"
80 A%=USR(5%)
```

А программа

```
10 IFPEEK(32%)=0%TH?CHR$(155%)
20 ?CHR$(12%)CHR$(148%)CHR$(158%)
      CHR$(145%)
30 POKE-50%,192%
40 POKE-50%,0%
50 IF INP(-50%,64%)TH50ELJ%=PEEK(-78%)
50 PSET(X%,Y%),2%
70 IFJ%=8%ANDX%>0%THX%=X%-1%
80 IFJ%=25%ANDX%<255%THX%=X%+1%
90 IFJ%=26%ANDY%>0%THY%=Y%-1%
100 IFJ%=27%ANDY%<239%THY%=Y%+1%
110 GOT50
```

позволяет прорисовывать линии курсорными клавишами.

Оказывается, можно прорисовывать линии не только четырьмя основными цветами палитры БК, но и произвольным их сочетанием:

```
10 CLS
20 FOI%=0%TO 239%
30 POKE138%,I%+1000%
40 LINE(0%,0%)-(255%,I%),0%
50 NEXI%
60 POKE138%,0%
```


Этот же автор рекомендует при случайном выходе в монитор БК возвращаться в Бейсик-систему без потери программы в ОЗУ с помощью команды

```
?S120170
```

а при случайном затирании текста программы командами NEW или DEL попытаться восстановить его в памяти БК командой

```
POKE&02024,&02026
```

Д. Шеховцов предлагает организовать плавное смещение экранной страницы подпрограммой

```
10 AX=16383Z
20 FOIX=0%T0255X
30 POKE-76%PEEK(-76%)+1X
40 FOJX=AXT0AX+63
50 POKEJX.0%
60 NEXJX
70 AX=JX
80 NEXIX
90 POKE-76%.728X
100 RET
```

Строка 90 может быть заменена строкой

```
90 ?CHRX(140%)CHRX(140%)
```

для обеспечения исходной установки ячейки 204, например для вывода сообщения в служебную строку. Естественно, это справедливо только для случая, когда служебная строка уже предварительно была погашена тем или иным способом.

В. Владимиров прислал две подпрограммы, анализирующие слово состояния дисплея (ССД).

Анализ бита 0:

```
Анализ бита 0:
10 POKE &01000,&0104034
20 POKE &01002,&010015
30 POKE &01004,&0207
40 DEFUSR=&01000
50 IFMIDY(BINR(USR(AX)),15,1)<>"0"TH
?CHRX(&0233)
```

При этом вывод информации будет происходить в формате 64 символа в строке независимо от режима показа, установленного ранее.

Анализ бита 3:

```
Анализ бита 3:
50 AX=BINR(USR(AX))
60 IFMIDY(AX,12,1)<>"1"TH?
"Включите регистр РУС"ELGOT80
70 GOT50
80 ?...продолжение программы
```

Программа продолжит свою работу только при соответствующем регистровом переключении.

Подобным же образом можно организовать анализ остальных битов ССД, что расширяет возможности программиста и дает интересные решения достаточно сложных проблем.

Читатель В. Константинов более двух лет успешно использует адаптированный им для БК алгоритм генератора случайных чисел, предложенный в книге Дж. Теннанта-Смита «Бейсик для статистиков».

```
10 W1=8192
20 W2=67101323
30 W3=W1*W1-W2
40 ?"Введите число из интервала
1-67101322 для задания
последовательности случайных
чисел:"
50 INPW
60 W4=INT(W/W1)
70 W=W3*W4+W1*(W-W1*W4)
80 IFW>W2 TH W=W-W2
90 X=INT(100*(W/W2)+1)
100 ?X:
110 GOT60
```

По мнению автора, этот алгоритм обладает рядом существенных преимуществ перед алгоритмом штатного генератора случайных чисел БК: большим рассеянием, отсутствием корреляции троек чисел, возможностью повторения любой из 67101322 последовательностей случайных чисел. Апробирован при разработках статистических моделей и программировании игр.

С. А. Паничев (Мурманская обл.) обратил внимание на ошибки, вкравшиеся в опубликованную в «ИНФО» (№ 4—90) схему БК-0010:

выводы 5, 3, 2 процессора D14 должны быть соединены в одной точке;

номера проводов в жгутах, к которым подключены выводы D14 — 1—16 (AD0—AD15), 22 (IAKO), 27 (INIT), 31 (SEL1), 32 (SEL2), 17 (SINC), 35 (DIN), 34 (DOUT), 20 (WTBT); выводы DMGO и BSY ни с чем не соединены;

на входы CAS микросхем DS9—DS16 должен подаваться сигнал CAS1 с микросхемы D19, т. е. они должны быть подключены к проводу 10 в жгуте;

на разъеме XT3 «МПИ» линия B21 ВЫВОД Н должна иметь номер 34.

И. ПАНЧЕНКОВ

О возможностях функции MID ☒ при работе с файлами на КУВТ-86

При написании программ на БК-0010-01, работающих с файлами данных (всевозможные справочники, некоторые обучающие и контролирующие программы и т. п.), возникает задача обращения к файлам, записанным на магнитные носители ДВК. Их имена нужно вводить в программу по мере необходимости с клавиатуры БК. Естественно присвоить введенное имя какой-либо символьной переменной:

```
10 INPUT "ВВЕДИТЕ ИМЯ ФАЙЛА":НХ
```

Однако если затем использовать оператор

```
20 OPEN "ТТ:"+MID$(НХ,1,6) FOR INPUT
```

или, что еще проще,

```
20 OPEN "ТТ:"+MID$(НХ,1) FOR INPUT
```

то это неизбежно приведет к ошибке 19 — ошибке устройства ввода-вывода, делающей невозможным чтение файла по имени, записанном в символьной переменной НХ. Можно ли избежать описанной ситуации? Можно, если формировать имя файла, используя функцию MID☒.

Известно, что длина имен файлов, читаемых с магнитного носителя ДВК, ограничивается шестью символами. С учетом этого запишем оператор открытия файла следующим образом:

```
20 OPEN "ТТ:НХ" FOR INPUT
```

При таком обращении к файлам вы будете застрахованы от ошибок устройства ввода-вывода, файл с требуемым именем успешно считывается в память БК.

85

ДОСКА ОБЪЯВЛЕНИЙ

- Предлагаю обмен программами для БК-0010. Имею более 300 программ.
142115, Подольск, ул. Филиппова, ба, кв. 174. С. А. Новиков
- Хотел бы переписываться с владельцами ПК «Ассистент».
700206, Ташкент, м-в Юнус-Абад, 14-30а-96. М. Р. Мун.
- Предлагаю новый компьютерный журнал для пользователей «Вектора-06Ц». Журнал представляет собой программу, записанную в формате начального загрузчика. Автор программ для «Вектора» приглашаю к сотрудничеству. Имею много программ для «Вектора», БК-0010.01, РК-86. Ваше письмо не останется без ответа!
446430, Самарская обл., г. Отрадный, Главпочтамт, до востребования. П. А. Желнов
- Приобретем ППС по любым предметам, инструментальные средства, игры для ПК-8000 «Сура» по безналичному расчету.
352765, ССР Адыгья, Майкопский р-н, ст. Кужорская, СШ № 12. Директор К. И. Лебедев
- БКашники из Челябинска, откликнитесь!
454126, Челябинск, ул. Образцова, 28, кв. 2. С. А. Панов
- В нашем училище успешно применяются в учебном процессе программные разработки для машинных курсов информатики на ПЭВМ «Корвет». Готовы обмениваться программами и переписываться.
215010, Смоленская обл., г. Гагарин, ул. Гагарина, 47, СПТУ-8. Б. А. Орлов
- Хочу обменять кассету с 64 игровыми программами (в машинных кодах, записаны с помощью прилагаемого HELP7R1) для БК-0010 на примерно такое же количество игр для «Спектрума».
301670, Тульская обл., Новомосковск, ул. Молодежная, 6, кв. 158. Е. В. Юдаев
- Хотел бы обмениваться программами и опытом работы на БК-0010. Готов поменяться и поделиться. Имею около 100 программ.
743500, КСР, Муынак, ул. Сборная, 103. К. К. Утегулов

- Хочу обмениваться программами для «Спектрума», «Дельты-С», «Карата» и других совместимых с ними.
301670, Тульская обл., г. Новомосковск, ул. Молодежная, 6, кв. 156.
К. Ю. Соколов.
- Обменяюсь программами и информацией для ПЭВМ «Вектор-06Ц». Имею более 250 программ.
456324, Миасс-24, а/я 139. Н. А. Данилов.
- Интересуют: цветная графика БК-0010.01, динамические эффекты, бегущая строка, спрайты, время в кадре, работа с дисководом, синхронизация с видеоманитофоном, видеокамерой, обмен программ.
342343, Донецкая обл., г. Докучаевск, ул. Лихолетова, 10, кв. 19.
Л. И. Степанов.
- Предлагаю для обмена большой набор программ для «Атари».
630075, Новосибирск, ул. Д. Ковальчук, 396, кв. 59. Сергей
- Готов обмениваться программами для БК-0010.01. Имею несколько сотен программ.
222120, Минская обл., Борисов-9, а/я 24. Ю. Г. Луцкевич
- Прошу откликнуться пользователей ПЭВМ «Поиск» (Киевского ПО «Электронмаш»), имеющих хоть какое-нибудь программное обеспечение на МЛ, в особенности пакет базового ПО.
270086, Одесса, ул. Заболотного, 55/12. В. В. Банных
- Хотел бы обмениваться информацией и программами с владельцами ПЭВМ «Микроша» и «Криста».
195176, Санкт-Петербург, ул. Б. Пороховская, 41, кв. 243. Тел.: 224-44-91. С. Ю. Головин
- Хочу переписываться и обмениваться программами для РК-86 и «Спектр-001». Умею переделывать программы «Микроши» для РК-86.
163040, Архангельск-40, а/я 53.
К. В. Ануфриев
- Имею более 500 программ для БК-0010.01. Предлагаю обмен. Прошу откликнуться тех владельцев БК, кто может оказать помощь в размножении каталога (написан в EDASP-10).
320125, Днепропетровск, ул. Червоного казачества, 26, кв. 56; тел.: 24-03-32. В. Ф. Рыжих
- Имею около 900 программ для БК-0010.01, в том числе много новых игровых. Высылаю каталог.
700107, Ташкент, ул. Фархадская, 7, кв. 11. П. В. Арзяев
- Прошу откликнуться владельцев компьютеров «Атари», желающих обмениваться программами.
692438, Приморский край, Дальнегорский р-н, пос. Краснореченск, ул. Октябрьская, 15, кв. 18.
П. О. Старокожев
- Хочу переписываться, обмениваться программами с владельцами ПК «Специалист». Имею около 150 программ.
310168, Харьков, ул. Героев труда, 12, кв. 147. В. А. Зарубин
- Хочу обмениваться программами с владельцами БК. Имею около 400 программ.
142115, Подольск, ул. Филиппова, 4, кв. 83. Д. В. Склянов
- Приглашаю владельцев компьютера «Вектор-06Ц» объединиться в клуб по переписке. Прошу присылать идеи и предложения. Мною создан синтезатор речи для «Вектора».
420066, Казань, а/я 169. Р. Р. Шаги-дуллин
- Хочу переписываться и обмениваться программами на Бейсике для КУВТ-86, ДВК-3, БК-0010.01.
374267, Азербайджанская ССР, г. Исмаиллы, ул. Октября, д. 15.
Д. А. Айдынов
- Начинающий пользователь «Синклер Спектрума» очень хотел бы получить игровые и системные программы.
354003, Сочи, ул. 60 лет ВЛКСМ, 8, кв. 69. Д. В. Моргунов
- Желающих обменяться программами для «Атари» и «Спектрума», прошу написать по адресу:
641544, Курганская обл., Мокроусовский р-н, с. Мало-Мостовое, С. М. Кокорину
- Хотел бы переписываться с владельцами компьютера «Вектор-06Ц». Имею более 50 программ в формате загрузчика.
690037, Владивосток, ул. Патриса Лумумбы, 66, кв. 16. Р. В. Любимов

Книга — почтой для пользователей «Электроника БК-0010».

Адрес: 105187, Москва, Кирпичная ул., 39—41, ПТО «Магистр-2», ИТС

**Для читателей, начавших выписывать «ИНФО»
только в этом году, приводим список статей о БК-0010,
опубликованных в течение 1991 г.**

Барсуков А. Подключение принтеров «Электроника МС-6312» и «Электроника МС-6313» к БК	
Барсуков А. БК-0010.01 в роли осциллографа	5
Барсуков А. Энергонезависимое ОЗУ БК-информ	2 4
<i>Информация о назначении контактов некоторых разъемов БК и функциях контактов клавиатуры</i>	
Гармашов А. Удобный аналог команды MERGE	2
Губаренко В. Волшебная сила двух ячеек	3
<i>Об использовании режима псевдоклавиатуры, позволяющего в автоматическом режиме исполнять директивы Бейсика</i>	
Диков А., Калашников А., Кулаков А. TermOS	2—3
<i>Описание операционной системы, использующей в качестве внешнего запоминающего устройства полностью управляемый БК магнитофон, и необходимых доработок магнитофона и компьютера</i>	
Еремин Е. Как работает команда MARK	6
Ермаков А. Программатор микросхем ППЗУ 558PP3 для БК-0010	3
Зальцман Ю. Архитектура и ассемблер БК	1—5
Зальцман Ю. Продолжительность исполнения команд на БК-0010	6
Зальцман Ю., Михайлов В. БК-0010 — речь и слух: возможности и реальность	2
Ингорь А. Конвертор Бейсик — ассемблер для БК	1
Ингорь А. Просмотрщик памяти	2
Калейдоскоп БК	1, 5
Карагиоз С. Многоформатный верификатор-каталогизатор	6
Ким В. Важные восьмеричные числа	3
Козлов О. «Куплю БК...»	4
Коренков В. Диспетчер ОЗУ и операционная система для БК-128К	6
<i>Описание схемы увеличения ОЗУ БК-0010 до 128К байт и программы, позволяющей использовать эту дополнительную память</i>	
Кошелев В. Подключение термопринтера 15ВВП180-002 к БК	6
Кумандин С., Соколов А. «Электронный диск» для БК-0010	1
Луцкевич Ю. 4УСЦТ — монитор	6
<i>Описание доработки телевизора 4УСЦТ, позволяющей с легкостью переключать его в режим монитора</i>	
Матвийчук С. Увеличение символов на экране	3
Михайлов П. Улучшение сопряжения БК и ТВ	5
Молдавский З. Работа с символьными величинами	4
<i>Оригинальный прием «обхода» ошибки компилятора Бейсика</i>	
Молчанов А. Оптимальный драйвер магнитофона для БК	3
Никонов П. Спрайты на БК	3
Панченков И. Кирпичики ваших программ	3
Парсаданов О., Янушко А. Защита программ от несанкционированного доступа	
Подгорецкий А. Дополнительные сведения о системном таймере	6
Рекомендации по оформлению программного обеспечения ПЭВМ БК-0010/0010.01	1
Рекомендации по подключению внешних устройств к порту ввода-вывода БК-0010	3
Садовников М., Петрук О. Упрощение организации движения изображений в Бейсике	5
Соколов А. Принципы работы БК-0010	1
<i>Описание работы микропроцессора K1801BM1A, системной шины, ОЗУ, устройств ввода-вывода и системы прерываний БК на «физическом» уровне</i>	
Сулейманов Т. и др. Цветной монитор как черно-белый	6
<i>Об использовании цветного монитора в одноцветном режиме</i>	
Фролкин Б. О «дребезге» клавиатуры БК-0010	6
Хорунжий В. Подключение принтера «Электроника МС-6312» к БК	4
<i>Приведен текст драйвера для управления МС-6312 из МИКРО8К</i>	
Чернолясов А. О символьных переменных	3

И. ГОЛИЦЫНА

НИИ ССО АПН СССР, г. Казань

88

Использование оболочки ЭС для создания ППС

Как показал наш опыт [1], с использованием оболочек экспертных систем возможности создания ППС на компьютерах профессионального назначения с большим объемом оперативной памяти расширяются. При этом, однако, предъявляются дополнительные требования к отбору содержания учебного материала для ППС и установлению внутрен-
них связей между отдельными его частями, увеличивается нагрузка на дидактическую и методическую сторону, и в целом процесс создания ППС поднимается на качественно более высокий уровень, отвечающий современному уровню развития информатики и вычислительной техники, новым информационным технологиям обучения.

Мы предлагаем ППС «Физические методы в астрономии», которое предназначено для использования в курсе «Физика и астрономия» в ССУЗ, на занятиях по физике и астрономии в средних школах, а также для использования на занятиях по соответствующим предметам в педагогических вузах. ППС представляет собой информационно-справочную обучающую систему, реализованную на ПЭВМ IBM PC и ЕС-1840.

ППС состоит из оболочки экспертной системы EXPERT, реализованной на языке Турбо-Паскаль [2], базы знаний и файла, содержащего графические подпрограммы.

Оболочка EXPERT предназначена для организации экспертной системы справочного назначения, которая предоставляет пользователю информацию об объекте, выбранном в качестве цели, после прохождения последовательности правил в базе знаний и заполнения базы данных после ответов на вопросы, конкретизирующие свойства объекта — цели. Текст программы EXPERT, опубликованный в [2], адаптирован нами для реализации обучающей системы. Основные изменения касаются процедур ввода ответа и вывода текстовой информации на экран — в оригинальной программе для ввода-вывода используется текстовый экран, мы пользуемся графическим. Оболочка дополнена модулями, ответственными за выведение на экран гра-

фической информации, дополняющей и расширяющей текстовую. С помощью графических подпрограмм на экран выводятся рисунки, чертежи, таблицы, формулы, расширяющие дидактические возможности системы.

Организация учебного материала в оболочке экспертной системы позволяет создать разветвленное дерево знаний и вопросов и обеспечить свободу передвижения обучаемого по каждой из его ветвей.

Базу знаний этой системы можно охарактеризовать как учебный модуль, содержащий учебный материал:

по физике (основные сведения о законах геометрической оптики, спектральном анализе, шкале электромагнитного излучения, химическом действии света, законах фотоэффекта и фотометрии);

по астрономии (информация о звездах, галактиках, Солнце, полученная из наблюдений в различных диапазонах длин волн с использованием различных методов наблюдений);

вопросы, ориентированные на применение знаний по физике в области астрономии и астрономического приборостроения.

В режиме «меню» учащимся предоставляется возможность выбрать:

астрономический объект для наблюдения (звезды, галактики, Солнце);

диапазон наблюдений (видимый, инфракрасный, радио-, гамма-, ультрафиолетовый, рентгеновский);

цель наблюдения выбранного объекта (определить химический состав, температуру, яркость, скорость движения, размер).

В зависимости от своего выбора или задания преподавателя учащийся может передвигаться по разным «веткам» дерева знаний. При общей свободе передвижения по дереву знаний структура элементарной ветки является жесткой, т. е. внутри такой ветки информация предъявляется в строго заданной последовательности и, как правило, содержит нестандартные вопросы, помогающие учащемуся самостоятельно осмыслить воз-

возможности применения физических понятий и законов в практической астрономии и астрофизике. Все ответы учащихся записываются в базу данных и по окончании работы могут быть выведены на экран.

Чтобы пояснить структуру организации учебного материала в базе знаний, приведем фрагмент базы знаний и пример работы экспертной системы с этим фрагментом. База знаний состоит из следующих компонентов:

а) описания возможных значений многозначных объектов, например:

VAL (объект)=звезды, галактики, Солнце

VAL (диапазон)=видимый, инфракрасный, ультрафиолетовый, радио, рентгеновский, гамма

VAL (a54)=закон отражения, закон преломления

б) вопросов, которые необходимо предъявить учащемуся для продвижения по «дереву» базы знаний, например:

question (объект)-Поработав с этой программой, вы узнаете, как наблюдают звезды, галактики, Солнце, как определяют их химический состав, яркость, температуру. Выберите интересующий вас объект:

question (диапазон)-Излучение от небесных тел можно наблюдать в разных диапазонах длин волн $l = c/v$, где c — скорость света, v — частота излучения. Человеческий глаз воспринимает излучение с 5000 \AA ($1 \text{ \AA} = 1 \cdot 10^{-10} \text{ м}$). Выберите диапазон наблюдений:

question (a53)- С помощью солнечных телескопов создается неподвижное изображение диска Солнца. Это достигается с помощью специальной системы зеркал — целостата. Солнечные телескопы бывают горизонтальные (на севере) и вертикальные (на юге).

question (a54)- Какой закон геометрической оптики используется при построении солнечного телескопа?

question (кп3)- Верно, угол падения равен углу отражения.

question (кн3)- Вы ответили неверно — используется закон отражения — угол падения равен углу отражения.

в) системы правил (в базе знаний ППС 120 правил), которая определяет возможность «продвижения» по дереву знаний при работе с программой, например:

```
rule 1: if объект=солнце &
диапазон=видимый &
a53=&
A54=закон отражения &
кп3=&
then
очс=.
```

```
rule 2: if объект=солнце &
```

```
диапазон=видимый &
a53=&
a54=закон преломления &
кн3=&
then
очс=.
```

Рассмотрим фрагмент работы ППС при «продвижении» учащегося по правилам 1 и 2.

В начале работы системы на экран выводится вопрос 1, дополненный пронумерованным списком значений «объект», который завершается комментарием: «Введите номер от 1 до 3». После того как учащийся выбирает значение «Солнце» вводом цифры 3, на экран выводится вопрос 2, дополненный списком значений объекта базы знаний «диапазон» и комментарием: «Введите номер от 1 до 6». Предъявление этого вопроса сопровождается выводом на экран схемы электромагнитной шкалы длин волн с указанием названий диапазонов и соответствующих им значений длин волн. Выбор диапазона «видимый» (нажатием цифры 1) приводит к дальнейшему продвижению по правилу 1. В этом случае на экран выводится вопрос 3, который заканчивается комментарием: «Нажмите ВВОД». После этого предъявляется вопрос 4, дополненный пронумерованным списком значений объекта «a54» и комментарием «Введите номер от 1 до 2». Если учащийся выбирает правильный ответ на вопрос — 1 (закон отражения), то в соответствии с правилом 1 ему предъявляется комментарий 5, в противном случае — по правилу 2 — комментарий 6. Вывод комментария сопровождается выводом на экран схемы установки горизонтального солнечного телескопа. В правилах 1 и 2 объект «очс» представляет собой промежуточную переменную, необходимую для дальнейшего «продвижения» по правилам в базе знаний.

Как можно понять из приведенного фрагмента, оболочка экспертной системы наилучшим образом подходит для создания информационно-справочной обучающей системы, т. е. ориентирована на предъявление вопросов с выборочным типом ответа, что существенно сужает возможности разработки с ее помощью диалоговых и контролирующих ППС. Возможность предъявления вопросов с конструируемым ответом при этом не исключается, однако анализ такого ответа требует сильного «разветвления» дерева правил, что приведет к неоптимальному разбуханию базы знаний.

Исходя из всего вышесказанного можно заключить, что для профессиональных типов ПЭВМ возможно использование существующих оболочек экспертных систем для построения педагогических программных средств. Это предоставляет большую свободу

в организации учебного материала и способах его представления, преподавателям предоставляется большая свобода выбора при работе с учебным материалом и возможность творчески подойти к организации занятий по соответствующим темам.

При этом повышаются требования к разработчикам ППС в следующих основных направлениях:

отбор содержания учебного материала для базы знаний экспертной системы и установление обширных внутренних связей между отдельными его частями требуют высокого профессионального уровня и компетентности разработчика, который должен быть экспертом в предметной области ППС;

увеличивающаяся нагрузка на дидактическую (создание обучающих модулей или

укрупненных дидактических единиц) и методическую (нестандартная организация учебного материала, поиск адекватных методов обучения с помощью ЭВМ) стороны создания ППС требует высокого уровня педагогической подготовки разработчика ППС;

для достижения свободы «передвижения» по «дереву знаний» необходима четкая организация учебного материала в базе знаний, что требует скоординированных действий программиста и эксперта.

Литература

1. Голицына И., Нарыкова И. Компьютер на уроке физики // Информатика и образование. 1990. № 3.

2. Сойер Б., Фостер Д. Л. Программирование экспертных систем на Паскале. М.: Мир, 1990.

Описанная обучающая система реализуется МАП «Бакалавр».

Предлагаются также следующие оригинальные по содержанию пакеты ППС для КУВТ «Ямаха», ПЭВМ БК-0010 (проводится адаптация для РС-совместимых компьютеров). Пакеты предназначены для средних общеобразовательных и профессиональных учебных заведений, педагогических институтов и системы переподготовки педагогических кадров.

«Химия-8» — 14 программ обучающе-контролирующего и игрового типов.

«Химия-9» — 17 программ исследовательского типа с элементами имитаций, модельных опытов и экспертных систем.

«Физика» — пакет из 9 программ для имитационного моделирования физических экспериментов и решения творческих конструкторских задач.

Обращаться по адресу: 420063, г. Казань, ул. Исаева, 12, НИИ ССО АПН СССР, МП «Бакалавр».

Оригинальные компьютерные игры не для развлечения, а для развития мышления

вам предлагает ПЕДАГОГИЧЕСКИЙ ЦЕНТР «ЭКСПЕРИМЕНТ»!

7 развивающих игр для дошкольников и младших школьников, которые создают условия для активизации умственной деятельности, способствуют формированию «умения учиться», развитию у детей творческого мышления, развивают алгоритмическое мышление.

Игры реализованы на ПЭВМ «Ямаха», БК-0010, IBM PC.

Стоимость комплекта — 4700 руб., отдельных игр — от 500 до 700 руб. (без стоимости носителей информации). Оплата наличными, по перечислению, наложенным платежом.

Наш адрес: 226063, г. Рига, а/я 90;
тел.: 597387, с 21 до 23 часов.

Новый вариант школьной программы

В 1989 г. в порядке эксперимента, я начал преподавание информатики в VI классе общеобразовательной средней школы с 11-летним циклом обучения.

В связи с этим возникла необходимость сформулировать цели обучения и указать способы их достижения. Таким образом была разработана программа курса обучения школьников информатике, о которой я хотел бы проинформировать своих коллег.

Программа рассчитана на 6-летний период обучения детей, начиная с 11-летнего возраста, но может быть легко адаптирована для 5-, 4-, 3- или 2-летних циклов.

Цели программы:

привить детям навыки работы с информационно-вычислительной техникой (ИВТ) и научить их самостоятельному применению ее для решения практических задач;

как можно раньше сформировать у детей стиль правильного мышления, развить его теоретическую направленность и воспитать информационную культуру;

создать предпосылки для широкого внедрения средств ИВТ в учебно-воспитательный процесс.

Программа курса, по которой я работаю, включает в себя несколько основополагающих тем современной информатики в такой последовательности:

1. Общее представление о вычислительной технике.

Раздел программы содержит сведения об использовании средств ИВТ в народном хозяйстве, электронно-вычислительных машинах и их составных частях, правилах работы с ЭВМ. Основное внимание уделяется практике работы школьников с компьютером, приобретению навыков общения с компьютером.

2. Языки программирования. Понятие об алгоритме и способах его записи.

Раздел содержит сведения о языках общения человека с компьютером, алгоритме и способах его записи. Основное содержание раздела составляет детальное изучение языка программирования Бейсик и практика работы с готовым программным обеспечением.

3. Основы логики и программирования.

Раздел программы включает краткое введение в логику, сведения о типах программ и принципах их конструирования. Основное внимание сконцентрировано на научении школьников чтению компьютерных программ и умению их тестировать.

4. Основные этапы решения задач обработки информации с помощью ЭВМ по схеме: объект—модель—алгоритм—программа—результат на примерах задач из школьных курсов.

Этот раздел учебной программы имеет исключительно экспериментально-практическую направленность и в полной мере дает школьникам представление о технологии решения различных задач средствами информатики.

5. Основы информационной культуры.

Даны сведения об информации и ее свойствах, о месте и роли информатики в системе научного мировоззрения, приемниках и передатчиках информации, способах ее хранения. Дается понятие автоматической обработки информации (АОИ).

6. Индивидуальная работа школьников.

Эта часть учебной программы предполагает использование средств ИВТ в целях изучения профилирующих учебных дисциплин или информатики по индивидуальным планам.

Объем программы по частям: 1, 2, 3, 4, 5 — по 31 академическому часу (всего — 155 академических часов); количество часов на раздел 6 устанавливается по усмотрению учителя.

После того как была сформулирована программа обучения детей информатике, нужно было подобрать учебную и методическую литературу, и в связи с этим мной был выполнен краткий методический разбор наиболее известных учебных пособий по информатике [1—7], сущность которого видна из табл. 1.

При анализе я использовал понятие методической формулы учебника, под которой понимаю порядок следования тем в учебном пособии. Методическая формула определяет, на мой взгляд, стратегию обучения предмету (термин «стратегия обучения» заимствован из публикации [9, с. 19]).

Для каждого учебного пособия в таблице даны: в первой строке — объем темы в процентах; во второй строке — объем упражнений по теме в процентах; в третьей строке — методическая формула учебника.

Характеристика тем:

А — ЭВМ;

Б — Языки программирования;

В — Алгоритмы и алгоритмический язык;

Г — Программирование;

Таблица 1

№ п/п	Характеристика тем					
	А	Б	В	Г	Д	Е
1	25	10	56	—	9	—
	20 В—А—Б—Д	13	67	—	—	—
2	10	—	69	—	15	6
	5 Э Е—В—А—Д	—	80	—	—	15
3	40	13	11	—	36	—
	35 А—В—Б—Д	26	13	—	26	—
4	12	26	46	Б	13	3
	6 В—Б, Г—Е—Д—А	28	57	Б	5	4
5	22	14	25	18	17	4
	9 Е—В—Б—Г—А—Д	15	37	21	13	5
6	16	—	50	—	29	5
	12 Е—В—А—Д	—	56	—	26	6
7	19	41	30	Б	5	5
	— Е—В—Б, Г—А—Д	—	—	Б	—	—
8	42	18	—	7	33	—
	43 А—Б—Г—Д	14	—	7	36	—

Д — АОИ;

Е — Информатика.

Выполненный анализ учебных пособий [1—7] показал их непригодность для решения поставленной передо мной задачи.

Что же не устраивало меня в этих учебниках?

1. Неясно выраженные цели обучения.
2. Предлагаемая стратегия обучения.
3. Отсутствие сборников упражнений и задач с ответами к ним.
4. Ориентация пособий в основном на обучение старшеклассников (явно запоздалое для этого возраста) алгоритмизации и программированию.
5. Ориентация на дефицитные или вообще отсутствующие в природе технические средства.
6. Отсутствие учебно-тренировочного программного сопровождения.
7. Низкая доступность изложения учебного материала в сравнении с переводными детскими компьютерными изданиями зарубежных авторов.

Необходимо отметить, что выбор стратегии обучения, заложенной в данной программе, был сделан не случайно, а на основе анализа ряда последних публикаций и личного опыта преподавания информатики. Так, например, авторы некоторых публикаций указывают на то, что психолого-педагогические,

дидактические и методические вопросы компьютеризации обучения разработаны недостаточно и еще ждут своего обстоятельного исследования [10, с. 15, 83]; авторы ряда публикаций [11—15] в разной степени критикуют постановку компьютерного всеобуча в средней школе, в частности официальную программу по основам информатики и вычислительной техники для старшеклассников [16, 17]; для моих же зарубежных коллег используемая мной стратегия обучения информатике не является новостью [8, 9] (в приведенной выше таблице дана методическая формула учебного пособия [8]).

Стратегия обучения школьников информатике, выраженная в изложенной программе, может быть описана короткой методической формулой, малоупотребительной в советских курсах ОИВТ: ЭВМ—язык—алгоритм—программа—АОИ—информационная культура.

Программа обучения, построенная по названной методической формуле, содержит, на мой взгляд, наиболее естественную последовательность подачи учебного материала.

Отсутствие подходящей для решения данной задачи учебно-методической литературы привело к необходимости разработки для каждого года обучения комплекта документации, сопровождающего курс, в состав которого входят:

- краткий конспект лекций;
- методическое пособие для учителя;
- методическое пособие для ученика;
- сборник листингов используемых программ.

К моему сожалению, календарно-тематический план и описание тактики решения методических задач обучения школьной информатике, изложенные в перечисленных документах, не смогут уместиться в столь короткой статье, являющейся всего лишь предисловием к этой работе.

Мое сообщение было бы неполным, если бы я не коснулся вопроса об оценке знаний учеников и о результатах эксперимента.

Для оценки знаний учащихся мной был применен объективный статистический метод, сущность которого заключается в следующем: ученику предлагается некоторое фиксированное количество элементарных заданий (т. е. таких заданий, которые можно однозначно оценить: выполнено или нет, уровень сложности этих заданий должен устанавливаться, исходя из состава изучаемого материала). Их полному и правильному выполнению соответствует «отлично». Оценка ученика определяется пропорционально количеству и качеству выполненных им заданий, при этом само собой разумеется, что

условия тестирования должны обеспечивать объективность результатов его работы. Итоговая оценка определяется как среднеарифметическое из оценок, полученных за весь период обучения.

Таким образом, по итогам первого года эксперимента были получены результаты, представленные в табл. 2.

Таблица 2

Классы	Количество учеников (в %)			
	«отлично»	«хорошо»	«удовлетворительно»	«неудовлетворительно»
VI	9	42	49	—
VII	12	66	20	2
VIII	25	55	18	2
IX	10	62	28	—

Предварительный отбор учеников в классы не производился. Занятия проводились с группами учащихся, состоящими в среднем из 15 человек, при средней обеспеченности ПЭВМ. Следует отметить, что в силу некоторой корреляции результатов оценочных работ представленные данные являются завышенными примерно на 11 %.

Пользуясь случаем, хочу выразить свою искреннюю благодарность Людмиле Гавриловне Якушиной, директору средней школы № 12 г. Подольска Московской области, оказавшей мне неоценимую помощь в проведении эксперимента и вернувшей мне возможность его продолжения в этом учебном году, а также коллективу методического кабинета Подольского горно, особенно Нине Николаевне Крейдиной, за внимание и поддержку в этот трудный для меня период работы.

Литература

1. Ершов А. П., Монахов В. М. и др. Основы информатики и вычислительной техники. Ч. 1, 2. М.: Просвещение, 1985—1986.
2. Ершов А. П., Кушниренко А. Г. и др. Основы информатики и вычислительной техники. М.: Просвещение, 1988.
3. Ваграменко Я. А., Антипов И. Н. и др. Электронно-вычислительная техника. М.: Просвещение, 1988.
4. Гейн А. Г., Житомирский В. Г. и др. Основы информатики и вычислительной техники. Свердловск: Изд-во Уральского ун-та, 1989.
5. Каймин В. А., Щеголев А. Г. и др. Основы информатики и вычислительной техники. М.: Просвещение, 1989.
6. Кушниренко А. Г., Лебедев Г. В. и др. Основы информатики и вычислительной техники. М.: Просвещение, 1990.
7. Шрайберг Я. Л., Гончаров М. В. Справочное руководство по основам информатики и вычислительной техники. М.: Финансы и статистика, СП Соваминко, Компьютерпресс, 1990.
8. Кершан Б., Новембер А., Стоун Д. Основы компьютерной грамотности. М.: Мир, 1989.
9. Вильямс Р., Маклин К. Компьютеры в школе. М.: Прогресс, 1988.
10. Гершунский Б. С. Компьютеризация в сфере образования: проблемы и перспективы. М.: Педагогика, 1987.
11. Тихонова А. Н. Еще раз о вычислительной технике // Коммунист. 1986. № 1. С. 113—114.
12. Радкевич И. О. О компьютере без дифирамбов // Лит. газ. 1986. 17 сент.
13. Эфрос Л. Б. ЭВМ в школе — реальность наших дней // ЭКО. 1984. № 11. С. 83—105.
14. Логвинов И. И., Логвинова Т. Л. Компьютерная грамотность в системе общего образования учащихся // Новые исследования в педагогических науках. 1985. № 2. С. 30—33.
15. Машбиц Е. И. Компьютеризация обучения: Проблемы и перспективы. М.: Педагогика, 1986.
16. Сундуков А. Это ли язык НТР? // Учит. газ. 1986. 3 апр.
17. Кривошеев В. Ф., Антипов И. Н., Боковнев О. А. Основы информатики — школьникам // Сов. педагогика. 1985. № 3. С. 12—16. ст 212, далее стр 89,

93

FROM дом TO работа STEP n

Первые десять парализованных пациентов в Кливлендском медицинском центре по оказанию помощи инвалидам используют компьютеры для управления своими мышцами. Экспериментальная электронная система функциональной электростимуляции (FES) состоит из датчиков и электродов, укрепляемых на те-

ЧТО?
МОЖЕТ
ЭВМ

ле, и компьютера, управляющего мышцами с помощью электрических импульсов. На пальце пациента укреплено простое устройство с двумя кнопками и ЖК-экраном. Первой кнопкой выбирается действие из отображаемого на экране меню («идти», «стоять», «подняться по лестнице» и т. п.), вторая прекращает действие.

Вряд ли это очень удобно, но по сравнению с неподвижностью...

Школьная электронная почта

Подходит к концу третий этап совместного советско-американского телекоммуникационного проекта «Школьная электронная почта». Он был задуман и проводится в качестве опережающего эксперимента, позволяющего оценить потенциальную роль, которую могут сыграть регулярные контакты между учебными заведениями, педагогами и школьниками различных стран.

СШ № 199 с углубленным изучением ряда предметов Черемушкинского района Москвы — один из участников этого эксперимента. Проект был рассчитан на три этапа, каждый продолжительностью в учебный год. На первом, подготовительном этапе (1988—1989) определялись школы, участники эксперимента знакомились друг с другом, устанавливалось коммуникационное оборудование.

Наша школа-партнер, в которой русский язык изучается как иностранный, находится в Нью-Йорке.

Летом 1989 г. в Москве был проведен двухнедельный семинар. Мы получили возможность лично узнать друг друга, обсудить планы совместной работы, познакомиться с имеющимся в мире опытом использования телекоммуникации в учебном процессе, методикой организации совместной работы.

В нашей школе, дополнительно к имеющемуся оборудованию, была установлена персональная ЭВМ типа IBM PC с модемом для передачи данных по коммутируемым каналам городской телефонной сети, а также люмафон, который позволяет передавать дополнительно к голосу статические изображения собеседников.

На семинаре мы совместно с координаторами проекта из американской школы выбрали тему проекта на второй год эксперимента (1989—1990).

По предложению американцев мы назвали совместный проект «Мусорное ведро моей семьи». Эта тема выбрана с учетом интересов школьников обеих стран. Она — всего лишь часть более обширной общей темы «Человек и большой город», сформулированной советскими учащимися. Почему именно мусорное ведро? Дело в том, что из домашнего мусора складываются огромные горы отходов в таких многомиллионных городах, как Москва и Нью-

Йорк, со всеми вытекающими отсюда последствиями.

Выбрав тему, мы разработали план поэтапного осуществления проекта. На первом этапе мы исследовали качественные и количественные характеристики отходов в доме каждого члена проекта с американской и советской сторон. Для эксперимента были взяты два дня: один — среди недели и второй — воскресенье. Мы смогли приблизительно узнать, сколько и какого мусора выбрасывается на городские свалки; кроме того, мы могли классифицировать мусор. Работа велась по единому плану, и результаты эксперимента можно было сопоставить.

В нашей школе был создан клуб «Электронная почта», члены которого, участники проекта, один раз в неделю собирались для обсуждения общих вопросов и четыре раза в неделю могли обращаться за консультациями к координаторам. Мы ввели балльную оценку различных видов работы, ежемесячно подсчитывали рейтинг учеников. На всех стадиях проекта мы обменивались информацией с партнерами из Нью-Йорка через компьютер. Но, кроме этого, мы имели возможность провести люмафонные конференции учеников наших школ. На первой конференции советские и американские школьники смогли не только услышать, но и увидеть друг друга, а также передать графические таблицы и диаграммы. Подобных конференций было еще две в течение учебного года, причем на апрельской конференции американские школьники уже из Москвы делились впечатлениями со своими родителями и учителями в Нью-Йорке.

На втором этапе работы в 1989—1990 гг. ребята наших школ выбрали несколько направлений для исследования. Члены клуба разделились на 6 групп, каждая из которых занималась разработкой своего проекта; темы исследований обсуждались с американскими партнерами при помощи электронных писем. Одна из групп, техническая, осуществляла компьютерное обслуживание проекта. Направления исследований были следующие: влияние твердых отходов на здоровье человека, биомассы как альтернативный источник энергии, утилизация твердых отходов, способы захоронения твердых отходов, влияние рационального питания на от-

ходы. В результате кропотливой работы были выявлены те методы и технологии переработки мусора, которые существуют в СССР и за рубежом.

В марте 1990 г. группа наших учащихся выехала в США для совместной работы над проектом. За две недели пребывания в Нью-Йорке ребята получили возможность обменяться собранной информацией, познакомиться с тем, как аналогичные вопросы решаются в США. Мы посетили городскую свалку Нью-Йорка и смогли воочию убедиться, как обстоят дела с захоронением отходов. Во время нашего визита в американской школе была проведена совместная конференция, на которой представители школ-партнеров доложили свои результаты. Кроме того, была организована встреча советских и американских школьников с руководителями проекта с обеих сторон. На этой встрече ребята ответили на вопросы, рассказали о сути проектов и поделились результатами своей работы.

В апреле 1990 г. мы встречали своих партнеров в Москве, дали им возможность посетить водоочистные сооружения нашего города. В Москве на совместной отчетной конференции были подведены итоги работы в 1989—1990 гг.

В этом учебном году с учетом пожеланий американских партнеров наши учащиеся выбрали следующие темы для работы: изучение степеней загрязненности воздуха некоторых районов Москвы выбросами промышленных предприятий, проблема кислотных дождей, транспортные отходы и состав атмосферного воздуха, проблема бродячих животных и грызунов в городе, архитектура как одна из причин стрессов у человека. Эти темы являются составляющими продолжающейся общей темы «Человек и большой город». В отличие от прошлого года проведение эксперимента было затруднено из-за сложности его постановки, и учащиеся могли получать сведения, проанализировав огромное количество литературы и побывав в различных организациях: СЭС, МИСИ им. Куйбышева, МХТИ им. Менделеева, МАДИ, Госкомприроде РСФСР, Экологическом комитете, Экологическом фонде, на различных предприятиях и т. д. Помощь в проведении эксперимента оказали сотрудники промышленно-экологического факультета МХТИ им. Д. И. Менделеева, подключив школьников к практике по мониторингу.

Итогом нашей работы было участие в экологической конференции, проведенной в апреле 1991 г. в г. Обнинске Обществом научно-технического творчества учащихся. Работы были доложены на секции «Эко-

логия» и напечатаны в сборнике, издающемся по итогам конференции.

Заключают нашу совместную деятельность взаимные визиты школьников.

Мы считаем работу над телекоммуникационными проектами удачной, и вот почему. В результате анонимного анкетирования мы выяснили, что причинами, приведшими учащихся в клуб, были интерес к экологии и возможность освоить компьютерную технику. На вопрос, получили ли они удовлетворение от работы, все ответили однозначно: «да». Особенно им импонировала самостоятельность в делах. Ребята отметили, что у них появились интерес к исследованиям, навыки планирования, умения корректировать свои действия с учетом мнения партнеров и т. д. У ребят стало сформировываться экологическое мировоззрение, чувство ответственности не только за себя, но и за окружающий мир. Занявшись вопросами чистоты города, они не могли не выйти на вопросы культуры, воспитанности человека. Это один из самых важных, с нашей точки зрения, аспектов воспитания. Никакая дидактика не сможет убедить человека сильнее, чем результаты его размышлений.

95

Метод проектов, навыки в планировании эксперимента, исследовательской деятельности, умение анализировать факты, делать выводы, излагать материал, выбирать критерии для классификаций и оценок и т. д.— все это необходимо для осмысленной и полноценной жизни человека. Чем раньше приходится сталкиваться с решением подобных задач, тем быстрее человек приобретает соответствующий опыт. Через такой проект во сто крат быстрее устраняется компьютерная безграмотность, так как освоение компьютера — необходимая составляющая работы и у человека создается целевая установка — освоить средство коммуникации. Ставя перед собой такую утилитарную задачу, учащиеся очень быстро понимают, что возможности использования компьютера значительно шире.

Совместный советско-американский проект — это мощный стимул для изучения английского языка. То, что школа с таким трудом вбивает школьнику годами, делает в мгновение ока послание через компьютер из США лично тебе, да еще не требующее отлагательства для ответа.

В наше время, когда остро стоит вопрос взаимопонимания между людьми, такого рода работа дает возможность лучше понять других людей, позволяет познакомиться с другим образом мыслей и жизни. А понимание — всегда путь к консолидации.



Компьютер — по почте!

Полупрофессиональный компьютер — в каждый дом!

Фирма «ЛИНТех» предлагает серийно выпускаемый, лучший в своем классе персональный компьютер ПК8010/20 «Корвет».

Параметры компьютера:

процессор КР580ИК80А, тактовая частота 2,5 МГц, быстродействие 625 тыс. операций в секунду;
объем ОЗУ 256К, из них 192К могут использоваться как электронный квазидиск;
графический адаптер 512×256 точек, 16 цветов или градаций яркости (лучше, чем адаптер GCA IBM PCI).

Имеются интерфейсы:

накопителя на гибких магнитных дисках (их может быть до четырех);
кассетного магнитофона;
CENTRONIX (принтера);
RS-232 и токовой петли;
локальной сети;
монохромного и цветного RGBI-монитора;
параллельного порта расширения системы.

Операционная система — CP/M-80 (дискетный вариант) или CAS-CP/M (для работы с кассетным магнитофоном вместо дискетов).

Программное обеспечение — самое разнообразное. Языки программирования Бейсик (полный стандарт MSX), Си, Паскаль, Ада, ПЛ/1, Фортран, ассемблер. Электронные таблицы SuperCalc и MultiPlan, СУБД dBASE II, великолепные текстовые и графические редакторы. Игровые программы, не уступающие играм для IBM PC по насыщенности графикой, цветовыми и звуковыми эффектами. Деловые и сервисные программы — на любой вкус.

Файловая совместимость с IBM PC с помощью специального программного обеспечения фирмы «ЛИНТех».

И, наконец, **уникальная возможность** работы со всеми программами даже на ПК в минимальной конфигурации — без дискетов. Ориентировочная цена такого комплекта 3000 рублей. В комплект входит пакет программ и система CAS—CP/M. Возможна оптовая поставка компьютеров. Учебным организациям поставляется КУВТ «Корвет» с обеспечением установки и пусконаладочных работ.

Мы гарантируем техническое обслуживание купленных компьютеров и сопровождение программного обеспечения.

Всем, кого заинтересовал компьютер или программное обеспечение фирмы «ЛИНТех», рекомендуем направить свой запрос по почте в наш адрес. Мы немедленно вышлем Вам каталог!

Всем организациям и частным лицам, имеющим персональный компьютер или КУВТ «Корвет», а также его аналоги («Нейва», «Форманта», «Квант-8» и др.),

фирма «ЛИНТех» предлагает: объединяйтесь!

Мы проводим регистрацию и создание единой базы данных о всех пользователях ПК «Корвет»! Каждый зарегистрированный пользователь имеет возможность получить бесплатно пакет игровых программ. Кроме того, Вы получаете уникальную возможность всегда быть в курсе всех новостей, узнать, где и на каких условиях можно приобрести интересующие вас программы и многое другое. Делайте первый шаг к созданию Ассоциации пользователей!

Вышлите в наш адрес следующую информацию:
название организации, фамилия, имя и отчество ответственного лица;
почтовый адрес и контактный телефон;
какими компьютерами, КУВТаами вы владеете;
для каких целей и задач они используются;
каким программным обеспечением (как официально, так и нет) вы пользуетесь;
какое программное обеспечение вас интересует.

Владельцам КУВТ «Корвет» мы предлагаем модернизацию класса с заменой головной машины на IBM-совместимую, что позволит на порядок улучшить сервис, увеличить дисковую память и многократно ускорить работу сети.

Мы готовы тиражировать любые интересные программные или аппаратные разработки для «Корвета», а также сотрудничать со всеми организациями, занимающимися тиражированием программных средств.

Наш адрес: 119501, Москва, а/я 942
Телефон: (095) 273-50-14
Телефакс: (095) 292-65-11 LINTEN Box 9318

ВНИМАНИЮ ЧИТАТЕЛЕЙ!

6—9 апреля 1992 г. в Москве состоится международная конференция Восток — Запад по новым информационным технологиям в образовании.

East—West Conference
on Emerging Computer Technologies in Education
Moscow, April 6—9, 1992

Организаторы конференции

Международный центр научной и технической информации (МЦНТИ), Советская ассоциация искусственного интеллекта (САИИ), Association for the Advancement of Computing in Education (AACE) (Международная ассоциация за прогресс в компьютерном образовании).

Цель конференции

Обмен идеями, теоретическими и практическими результатами, установление прямых контактов между специалистами «запада» (США, Западной Европы) и «востока» (СССР, Восточной Европы, стран «третьего мира»). Знакомство участников с современным состоянием разработок и исследований по применению новых информационных технологий в образовании.

В программе конференции предусмотрены пленарные и стендовые доклады, дискуссии, демонстрации программного обеспечения, а также лекции ведущих западных и советских специалистов в области компьютерного образования. Конференция будет сопровождаться выставкой аппаратных и программных средств, применяемых в учебных целях. Конференция и выставка пройдут в здании МЦНТИ в Москве.

Тематика конференции

На конференции планируется рассмотреть актуальные вопросы применения новых информационных технологий в образовании по следующим трем направлениям:

- применение искусственного интеллекта в компьютерном обучении,
- учебные гиперсредовые и многосредовые системы (hypermedia and multimedia),
- учебные среды и «микроміры», учебные моделирующие системы.

Адрес оргкомитета:

125252, Москва, ул. Куусинена, 21 б. МЦНТИ,
Конференция «НИТ в образовании», Брусиловскому П. Л.
Электронная почта: eastwestapl@icsti.su
Телекс: 411925 МСНТИ
Факс: 942 00 89
Телефоны: 198 74 81, 198 73 20, 198 70 55.

В. ЦИКОЗА

Альтернативная задача II тура

В 1991 г. на IV Всесоюзной олимпиаде школьников по информатике в г. Бишкеке была предпринята попытка преодолеть засилье «традиционной математики» — вычислительные и *P*-полные переборные задачи по геометрии и комбинаторике, на рутинность которых постоянно ропщут сами участники.

На II тур были вынесены две примерно равносложные задачи по «нетрадиционной информатике»: о распознавании неоднозначности декодирования информации и об оптимизации программ. Жюри выбрало первую из них, опасаясь неизвестности и непривычности для школьников проблематики второй задачи.

По мнению же автора, именно незнакомство школьников с методикой оптимизации делает вторую задачу особенно интересной, в известной мере нивелируя различный уровень подготовки (как мы увидим, при решении задачи практически не требуется знания математики и стандартных алгоритмов) и выводя на первое место умение думать, смотреть и видеть, ну и, конечно, программировать.

Знакомство с одним из подходов к решению таких задач будет, несомненно, полезно будущим участникам олимпиад.

Приведем условие задачи.

Дано описание некоторого алгоритмического языка.

Требуется написать программу-оптимизатор, которая вводит текст программы на данном языке, проводит в ней перечисленные ниже преобразования и выдает в качестве

данные: число — последовательность десятичных цифр без знака
 переменная — большая буква латинского алфавита (*A ... Z*)
 выражение — объект, либо объект+объект, либо объект-объект
 условие — объект=объект, либо объект>объект

операторы:

присваивание: *переменная:=*

переменная:=

выражение

X:=Y+7

вывод: *PRINT*

PRINT

выражение

PRINT X

ветвление: *IF*

IF

условие

IF 0>A

оператор

A:=0

...

PRINT 0

ELSE

оператор

] может
отсутствовать

ELSE

PRINT A

ENDIF

ENDIF

цикл

WHILE

условие

WHILE A>0

оператор

A:=A-1

...

END

END

Операторы располагаются на отдельных строчках без разделителей, как указано выше; пробелы игнорируются.

результата текст получившейся программы:

а) свертка констант: выполнение операций

над константами и подстановка вместо переменных определенных значений, например:

$X := 5 + 2$ можно заменить на $X := 7$,

$Y := 7 + X$ $Y := 14$

б) удаление присваиваний несущественных значений (не используемых впоследствии в выражениях и условиях):

$A := 5$ — это присваивание несущественно (можно удалить),

$A := 7 + B$, т. к. присвоенное значение не используется;

в) удаление неисполняемых циклов и фрагментов ветвлений, например:

$IF\ 7 > 5$ можно заменить на $A := X$

$A := X$ так как условие всегда

$ELSE$ истинно

$A := Y$

$ENDIF$

г) удаление операторов, не влияющих на вычисление выводимых значений;

д) выдача списка неинициализированных переменных;

е) полная интерпретация программы, если она возможна.

Контроль корректности вводимой программы не проводить; считать, что все числовые константы в ней — от 0 до 9 (однозначные).

О чем эта задача.

Специфика задачи в том, что ее исходными и выходными данными является *текст программы* на довольно простом языке, а рабочими данными — программа, описанная этим текстом. *Оптимизацией* называется выполнение некоторых заданных преобразований с текстом исходной программы, в результате которых получается текст другой программы, *функционально эквивалентной* исходной (выдающей те же выходные данные при исполнении), но с улучшенными характеристиками (объем, скорость исполнения, количество вычислений и др.). *Полноту* оптимизации можно определить как невозможность выполнить те же преобразования в выходной программе применением любого другого алгоритма (например, человека), а *универсальность* оптимизатора — как способность его проводить оптимизацию для любого корректного входного текста.

Как решаются задачи оптимизации.

Эффективность решений задач трансляции и оптимизации программ обычно оценивается общим числом проходов по тексту исходной программы и всевозможным ее внутренним представлениям.

Если решение неоднозначно (оптимизация и вывод результата по ходу чтения исходной программы невозможны), то исход-

ную программу обычно переводят во *внутреннее представление*: чтобы не повторять синтаксический разбор текста, сохранять информацию, выработанную на предыдущих проходах, экономить память и просто для удобства обработки. Это представление называют *управляющим графом*, хотя обычно достаточно частного вида графа — дерева.

Использование графа программы приводит к двум вспомогательным проходам по программе: для построения графа по тексту и для выдачи текста выходной программы. Оптимизируется же не текст, а граф внутреннего представления, что технически удобнее.

Традиционные методы оптимизации опираются на специальные представления операторов и выражений (например, в виде триад или тетрад [1]), которые позволяют эффективно проводить такие виды оптимизации, как исключение ненужных вычислений и неисполняемых фрагментов, сокращение вычислений путем использования ранее вычисленных общих подвыражений, вынесение инвариантных вычислений из циклов и т. д.

Общедоступной литературы по оптимизации, к сожалению, мало. Дополнительную информацию следует искать в книгах по трансляции языков программирования и построению компиляторов, в задачи которых составной частью входит оптимизация (например, [1]).

Какое решение задачи ожидалось.

В условии задачи намеренно снижены требования к несущественным для основной цели элементам решения:

не требуется полнота оптимизации: она ограничена лишь заданными преобразованиями, причем для верного решения достаточно правильно обрабатывать частные случаи; допускается множество эвристик;

не требуется контроль корректности исходной программы;

допускается решение путем интерпретации программы.

Задача допускает полное однопроходное решение для программ без циклов, наличие которых делает алгоритм нелинейным, и полнота решения становится серьезной исследовательской проблемой. Достаточной считалась некоторая «матрешка эвристик», в основном в рамках однопроходного решения. Схемой оценки предусматривались и иные эвристики (например, полная интерпретация), однако особо требовались обоснование выбранных алгоритмов, анализ их полноты и границ применимости, позволяющие судить о понимании школьниками проблемы и видение частных случаев, из которых складывается решение.

Авторское решение задачи

Ниже рассматривается не полный, а сокращенный алгоритм оптимизации за один проход по исходной программе. Классические модели данных и методы оптимизации читатель может найти в [1].

Управляющий граф программы

Исходную программу удобно представлять в виде управляющего графа, узлы которого содержат всю нужную информацию о каждом ее операторе, а дуги задают порядок их исполнения. Например, для операторов присваивания и вывода следующий оператор определяется однозначно; оператор ветвления порождает два пути исполнения, сходящихся, впрочем, через некоторое время в одном узле; цикл можно представить как ветвление, в котором одна ветвь пустая (не содержит узлов-операторов), а другая циклически замыкается на заголовок цикла. Обратите внимание, что в языке допустима вложенность циклов и ветвлений!

Выбор модели данных для представления графа и кодирование программы в граф — задачи чисто технические и оставляются читателю, как упражнение.

Признаки состояния переменных

Определим несколько признаков состояний переменных в исходной программе, которые потребуются для оптимизации.

Оператор присваивания делает переменную в левой части *определенной* если выражение в правой части содержит только константы и другие определенные переменные, и, соответственно, *неопределенной*, если в выражении есть неопределенная переменная. Состояние «определенности» сохраняется во времени до следующего присваивания и может произвольно меняться.

Такая формулировка «определенности» отражает только упрощенный способ, которым мы будем определять значение переменных при оптимизации программы: например, ею не охватываются переменные, значения которых неявно задаются циклами:

WHILE A>0

A:=A-1

END на выходе A=0

Следует отдельно проверять присваивания вида $X:=Y-Y$ (X определен).

Назовем переменную *инициализированной*, если она не используется в выражениях и условиях до присваивания ей первого определенного значения в программе.

Назовем переменную, ее значение и оператор присваивания этого значения *существенными* в данной точке программы, если это значение было хоть раз использовано

в выражении или условии после его присваивания. К несущественным присваиваниям следует отнести и те, которые присваивают то же самое определенное значение, а также вида $X:=X$ или $X:=X+0$.

Таблица состояний переменных

Поскольку переменные — однобуквенные и их не более 26, удобно завести сводную таблицу состояний переменных и предусмотреть для каждой переменной такие признаки:

значение — текущее значение или прочерк, если оно неопределенно;

присвоено — ссылка на оператор присваивания этого значения (узел в графе) — используется только для несущественных переменных;

существенна — «да/нет» — было ли использование этого значения;

инициализирована — «да/нет» — была ли инициализация переменной.

В дальнейшем признаки переменных определяются по текущим значениям этих полей

Элементарные преобразования

Требуемые в условии преобразования могут быть получены комбинацией следующих *элементарных преобразований*:

1) Вычисление константного выражения: $7+7=>14$.

2) Подстановка вместо переменной ее значения: $X=>6$.

3) Удаление оператора присваивания.

4) Выделение одной из ветвей оператора ветвления в последовательность операторов (вычисление условия и удаление ветвления).

5) Удаление оператора цикла со всем его содержимым.

6) Удаление оператора, не влияющего на выводимые результаты.

Достаточными условиями для выполнения элементарных преобразований являются, соответственно:

1) *Выражение* имеет вид *число*, либо *число+число*, либо *число-число*, условие имеет вид *число=число* либо *число>число*.

2) Значение переменной в текущей точке программы определено.

3) Значение переменной несущественно вплоть до следующего присваивания ей.

4) Условие в ветвлении тождественно ложно или тождественно истинно, т. е. не содержит переменных после элементарных преобразований 1—3.

5) Условие в цикле тождественно ложно. Тождественная истинность означает закливание — в этой ситуации логично прекра-

титель оптимизацию и выдать сообщение об ошибке.

б) Будет рассмотрено позже.

Оптимизация выражений

Для оптимизации выражений и условий используем вспомогательный алгоритм:

АЛГОРИТМ А

1) Если операнды выражения или условия — константы или определенные переменные, выполнить указанную операцию и занести результат в граф программы вместо данного выражения.

2) Неопределенные операнды — переменные сделать существенными: запомнить факт использования их значения.

3) Для решения пункта е): при обнаружении неинициализированной переменной выдать ошибку и прекратить работу.

Оптимизация линейных участков

Допустим, что входная программа линейна, т. е. не содержит ветвлений и циклов:

АЛГОРИТМ Б

1) Инициализировать таблицу переменных: все они не определены, не инициализированы и существенны (последнее позволяет не инициализировать ссылки на присваивания).

2) Обработать поочередно все операторы в порядке их записи до конца программы — по отдельному подалгоритму для каждого вида оператора.

3) В конце работы для всех несущественных переменных удалить последние операторы присваивания (на них указывают ссылки «присвоено»).

Подалгоритм обработки оператора присваивания

П1) Определить *целевую* переменную (ту, которой присваивается значение), два объекта-операнда и операцию (случай, когда операнд один, разобрать отдельно).

П2) Выполнить Алгоритм А для выражения справа от знака присваивания:

если выражение вычислено полностью, то его значение сделать значением целевой переменной, тем самым определив ее;

если в выражении осталась неопределенная переменная, то сделать неопределенной и целевую переменную;

особо обработать и удалить несущественные присваивания, сводящиеся к текущему значению ($X := X$, $X := X + 0$, $X := X - 0$). Завершить выполнение подалгоритма.

П3) Если целевая переменная несущественна, то удалить оператор присваивания ей несущественного значения (по ссылке «присвоено»).

П4) Ссылку «присвоено» целевой переменной настроить на текущий оператор: для доступа к нему при возможном удалении.

П5) Признак «существенно» целевой переменной установить в «нет»: начать отсчет использования нового значения.

П6) Признак «инициализировано» целевой переменной установить в «да», если выполнены все следующие условия:

переменная не инициализирована (первое присваивание);

переменная не существенна (не использована до инициализации);

присваивается константа (значение становится определенным).

Иначе состояние признака не менять.

Подалгоритм для обработки оператора вывода построите из Алгоритма А самостоятельно (следует учесть вывод значений — для пункта е)!).

Оптимизация ветвлений

Разбор ветвлений можно вести на том же проходе, после выполнения элементарных преобразований 1—3.

Условие не тождественно, если в него входит неопределенная переменная. В этом случае нельзя удалять ни одну ветвь и требуется раздельная оптимизация каждой из них с учетом контекста.

Проведем последовательный рекурсивный анализ каждой из ветвей по Алгоритму Б.

На вход подается одинаковая начальная таблица состояний переменных, а две выходные таблицы сравниваются при продолжении разбора. Это позволяет обнаружить неизменившиеся или одинаково изменившиеся состояния переменных и сохранить их.

Подалгоритм обработки ветвлений для Алгоритма Б

В1) Выполнить Алгоритм А для условия в заголовке оператора.

В2) Если условие сводится к константам, то оно тождественно, истинно или ложно: в этом случае удалить заголовок ветвления, строки *ELSE*, *ENDIF* и все операторы неисполняемой ветви. Продолжить Алгоритм Б по операторам оставшейся ветви.

В3) Если условие содержит существенные переменные, то для каждой из двух ветвей (даже если одна пустая и опущена!) создать локальную копию глобальной таблицы переменных* и рекурсивно выполнить шаг 2 Алгоритма Б.

В4) Для каждой переменной сравнить два ее локальных состояния и исправить глобальное:

а) удалить как несущественное последнее присваивание переменной перед ветвлением (по глобальной ссылке «присвоено»), если эта переменная не существенна в начале ветвления (глобальный признак «существенна» = «нет») и инициализирована в каждой ветви (локальный признак «инициализирована» = «да»);

б) значение: = «не определено», как подготовка к следующему шагу;

в) если локальные выходные значения определены, совпадают и несущественны, то удалить оба последних оператора присваивания по локальным ссылкам «присвоено» и вставить оператор присваивания данного значения после ветвления — для разбора на следующем шаге Алгоритма Б;

г) «существенна» = «да», если переменная су-

* При рекурсивном применении Алгоритма Б в локальной таблице переменных следует предварительно для всех переменных установить «существенна» = «да» (это блокирует удаление присваиваний перед ветвлением при разборе ветвей: они могут быть существенны для другой ветви!), а «инициализирована» = «нет» (это позволит обнаружить несущественность таких присваиваний на шаге В4а).

щественна по окончании разбора хотя бы в одной ветви, иначе не менять;

д) «инициализирована»: «да», если переменная инициализируется в обеих ветвях, иначе не менять;

Отложенное удаление присваиваний выходит за рамки однопроходного разбора, но наличие графа программы скрывает технические трудности «возврата к прошлому».

Оптимизация циклов

Полная оптимизация цикла по аналогии с ветвлением в общем случае невозможна; число вариантов исполнения неизвестно. Рассмотрение частных случаев приводит к неплохим эвристикам, но здесь ограничимся лишь локальной оптимизацией тела цикла и его контекста в рамках Алгоритма Б.

Заменим цикл ветвлением, одна ветвь которого пуста (цикл не выполняется ни разу), а другая содержит операторы тела цикла. Поскольку при исполнении цикла любой оператор в нем может предшествовать любому, все преобразования, использующие предшествующий циклу контекст, некорректны.

Локальную оптимизацию цикла и определение состояний переменных на выходе из него можно провести с одним проходом по телу цикла.

Учет влияния друг на друга тела и окружения цикла, а также конца цикла на начало может быть реализован более аккуратным разбором всевозможных сочетаний состояний переменных и еще одним проходом по телу цикла. Это легко увидеть, если развернуть тело цикла в линейную программу из N одинаковых фрагментов.

Подалгоритм обработки циклов для Алгоритма Б
Ц1) Выполнить Алгоритм А для условия в заголовке цикла.

Ц2) Если условие сводится к константам, то оно тождественно истинно или ложно: в первом случае — выдать ошибку, во втором — удалить весь цикл и завершить подалгоритм.

Ц3) Создать локальную копию таблицы имен, инициализировать ее, как в шаге 1 Алгоритма Б*; рекурсивно выполнить шаг 2 Алгоритма Б для тела цикла.

Ц4) Сравнить признаки переменных в глобальной и локальной таблицах переменных и внести изменения в глобальную:

а) сделать переменную неопределенной, если ее локальное и глобальное значения не совпадают;

б) если «глобально» переменная не существует, перенести в глобальную таблицу ее локальное состояние существования;

в) переменные, которые изменялись в цикле

* Очистка локальной таблицы переменных позволяет во имя корректности сведения «забыть» предшествующий циклу контекст — при этом теряются все возможности оптимизации на границах.

(это можно узнать сравнением ссылок «присвоенно»), сделать существенным, чтобы не удалять в дальнейшем присваивания из цикла;

г) если ни одна из переменных, входящих в оптимизированное условие цикла, не изменялась в теле, выдать сообщение о заикливании и прекратить работу.

Дополнительная эвристика решения задачи: вести оптимизацию по ходу интерпретации программы (сведение к подзадаче е)); при этом доступна более точная информация о переменных, ведется полная прокрутка циклов и оптимизировать можно полнее.

Как нетрудно видеть, Алгоритм А решает подзадачу а), а полный Алгоритм Б — б) и в). В конце работы Алгоритма Б просмотром полей «инициализировано» можно решить подзадачу д). Если неинициализированные переменные есть, то подзадачу е) выполнить невозможно, иначе она решается на проходе оптимизации (после преобразований) или отдельным проходом по оптимизированному графу.

Исключение несущественных вычислений для полной задачи

Решение подзадачи г) не может быть получено по ходу преобразований вида 1—5 и требует отдельного прохода по программе. Причем обратного.

Введем для каждой переменной еще один признак — «выводима», указывающий на значимость переменной для вычисления выводимых результатов. Можно заметить аналогию свойства выводимости со свойством определенности переменной:

оператор вывода делает присваивание константы переменной выводимой; выводит переменную определенной;

в присваивании свойство выводимости переходит от переменной в левой части к переменным в правой; определенность переходит из левой части в правую;

свойство выводимости распространяется от операторов вывода вверх по программе; определенность распространяется вниз от присваиваний;

распространение выводимости оканчивается присваиванием константы (выводимого значения); распространение определенности оканчивается присваиванием неопределенного значения;

Эти соображения объясняют необходимость обратного прохода по программе, точнее — обращения последовательности вычислений. Алгоритм для подзадачи г) может быть получен реверсированием Алгоритма Б с заменой терминов и действий по приведенной выше аналогии.

Обратим внимание лишь на некоторые особенности реверсирования:

1) Для большей полноты реверсировать следует уже оптимизированный граф программы.

2) Удалению подлежат только присваивания «невыводимых» значений и ставшие пустыми вследствие этого ветви и циклы.

3) Ветвления и циклы следует обрабатывать рекурсивно, причем:

переменные в условии делать выводимыми; для ветвления разбирать обе ветви и сравнивать в конце признаки выводимости;

условие ветвления разбирать после обеих ветвей, а условие цикла — до разбора его тела (почему?);

перед рекурсивным проходом цикла все локальные состояния переменных делать выводимыми или делать второй проход по циклу (см. замечание к подалгоритму Б для циклов).

Обоснование и детализацию реверсивного алгоритма проведите самостоятельно.

Проблемы к самостоятельному исследованию любознательному читателю:

1. Как исправить Алгоритм Б, чтобы удалить присваивания вида $X:=Y$? Какие изменения это вызовет в реверсивном алгоритме?

2. Как выполнять оптимизацию несущественных вычислений в выражениях? Пример:
 $Y:=Y+X$
 $Y:=Y-X$ } все вместе несущественно.

3. Как исправить Алгоритм Б, чтобы инвариантные присваивания, где можно, выносились из циклов и ветвлений?

Пример:

```
WHILE A>7  
K:=0  
A:=A-1  
ENDIF
```



```
K:=0  
WHILE A>7  
A:=A-1  
ENVIF
```

4. Существенно ли для реверсивного алгоритма, что граф программы оптимизирован преобразованиями 1-5?

Литература

1. Грис Д. Проектирование компиляторов для цифровых вычислительных машин. М.: Мир, 1975.
2. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. М.: Мир, 1978.

103

ОПЯТЬ!!! Опять эти несносные мальчишки! Ну что им еще надо? НЕ ЗНАЕТЕ?

А мы уверены, что ответ на этот вопрос дадут программы компьютерной диагностики, предлагаемые ПЕДАГОГИЧЕСКИМ ЦЕНТРОМ «ЭКСПЕРИМЕНТ»:

- Комплекс групповых методик диагностики развития школьников (для 5—6, 6—9, 9—18 лет).
- Методики исследования профессиональных интересов, склонностей, мотивов профдеятельности (для 14—16 лет).
- Методики исследования черт характера и темперамента, устойчивости нервных процессов.
- Методики оценки психологического климата в коллективе.
- Методики исследования личностных свойств.

Наш адрес: 226063, г. Рига-63, а/я 90;
тел.: 59-73-87, с 21 до 23 часов.

Транзисторный экран

В одном из выпусков «Что может ЭВМ» мы писали о появлении новой технологии изготовления активных жидкокристаллических дисплеев, которая позволяла увеличить скорость работы «неповоротливых» жидких кристаллов и уменьшить их размеры.

Не прошло и года, как Seiko Epson начала продавать лэптопы, основной особенностью которых стал многоцветный активный ЖК-дисплей, дающий столь качественное изображение, что появилась возможность создавать на его основе 32-битные рабочие станции. На десятидюймовом поле экрана, состоящем из 640×480 точек, 64-цветное полутоновое изображение разрабатываемой детали выглядит вполне естественно.

Остальные параметры компактного компьютера тоже в норме: 100 МВ жесткий диск, процессор 80386DX с частотой до 20 МГц, 2 Мбайт ОЗУ.

Телевизор с искусственным интеллектом

Что означают буквы JVC, знают, наверное, все. Высококласная бытовая техника, выпускаемая на предприятиях этой японской фирмы, не просто соответствует уровню мировых стандартов, но и определяет его. И неудивительно, что именно JVC еще раз подняла планку.

Здесь разработан пульт дистанционного управления телевизором, оснащенный элементами искусственного интеллекта. У владельцев таких видеосистем посте-

ленно появляется ощущение, что телевизор приспосабливается к их привычкам.

Микрокомпьютер, встроенный в новую видеосистему, анализирует и запоминает самые популярные передачи в течение суток, выделяя три наиболее часто используемых телеканала. Владельцу достаточно нажать кнопку «Любимые передачи» — и на экране, разбитом на несколько полей, в уменьшенном размере появятся передачи, идущие по этим каналам, и меню, позволяющее выбрать нажатием соответствующей цифры любой из них.

Кроме того, система может запомнить и обеспечить быстрый просмотр еще 25 наиболее популярных каналов, которые разбиваются на пять тематических групп: спутниковые, киновидеофильмы, спорт, новости и музыка.

Аналогичным образом система следит и за уровнем громкости. Сутки в ее понимании разбиты на 24 часа, для каждого устанавливается определенная громкость.

Если у владельца этой «мудрой» системы есть дети, он может ограничить их ежедневные телепорции определенным часом, после наступления которого пропадает звук, а изображение сменяется приятным голубым полем. Но при определенном упорстве с пульта дистанционного управления можно подобрать трехзначный цифровой пароль, который позволит смотреть передачи и после наступления «тихого часа».

Новая система сохранила достоинства предыдущих поколений подобных устройств. С ее помощью можно разбить экран на девять равных полей, в каждом из которых будут идти передачи выбранных заранее каналов; или организовать микроэкран, расположенный в нижней части основного экрана, с помощью которого можно контролировать события, происходящие в другой передаче; или разбить экран на два разных поля, в каждом из которых будет изображение, полученное от двух разных источников видеосигнала. Кроме того, видеопроцессор системы позволя-

ет осуществить «заморозку» удачного кадра и цифровое стробирование динамического изображения, при котором экран разбивается на девять областей, в каждой из которых фиксируется очередная фаза движения.

Телефакс или телеприемник?

Недостаточно быстрый обмен информацией в наше время уже не только неприятный факт, но и довольно убыточное дело. Поэтому руководители фирм не жалеют средств на сетевую поддержку компьютеров и телефаксы. Но здесь их поджидает второе препятствие, преодолеть которое даже с помощью денег удастся не всегда, — ограниченная пропускная способность телефонных линий.

Японская фирма Mitsubishi решила эту проблему, использовав для передачи информации разрешенный радиоканал. А для работы с таким каналом создала гибридный телевизор и факса, который принимает сигнал и, обработав его, печатает факсограмму на бумаге.

Радиофакс оборудован видеопамятью емкостью в 8 Мбайт, жидкокристаллическим дисплеем с высокой разрешающей способностью, который позволяет отобразить переданное изображение с помощью более чем полутора миллионов точек, и спецпроцессором, который позволяет осуществить обработку принятого сигнала и управляет работой всего комплекса.

Радиофакс может работать в двух режимах — напоминания в максимальном темпе, при котором в памяти устройства фиксируется до двух страниц данных для последующей обработки, или непосредственной печати с эфиром, когда факс может воспроизвести на бумаге страницу формата А4 за 8 с.

И хотя практическое использование радиофаксов в Японии началось не так давно — летом 1991 г., уже сейчас возникает вопрос: надолго ли хватит емкости эфира?

Реалистические изображения

Алгоритмы позволяют строить трехмерные изображения фигур в пространстве линейной перспективы (далее ЛП). Наиболее наглядным применением алгоритмов может быть создание картинок для мультипликации или игр.

ЛП бывает одно- и многофокусной. Многофокусная ЛП полностью задает пространство, т. е. дает возможность переносом фокусов размещать фигуру в любом месте и положении (с любым углом поворота). Но работа с такой ЛП неэффективна из-за сложности алгоритмов. Предлагается использовать однофокусную ЛП, но она не позволяет изменять положения фигуры. Для реализации в однофокусной ЛП возможности полного задания пространства необходимо ввести операцию поворота.

Входные данные. Фигура — объект в Декартовом пространстве, заданный по координатам. Прimitivesом фигуры является любой объект (точка, прямоугольник, линия и др.), который можно задать по координатам. Например, линия запоминается, обрабатывается и строится по координатам начальной и

конечной точек. Для данных алгоритмов тип примитива не имеет значения, поскольку в процессе обработки не обязательно знать, к какому примитиву относится данный список координат.

Фигура хранится в двухмерном массиве координат. Для экономии памяти и времени (на машинах без сопроцессора) можно сделать массив целочисленным, а для проведения вычислений с точностью до нескольких знаков после запятой, — записать в него координату, умноженную на 10, 100 или на другой масштабный множитель.

...	x1	x2	x3	...
...	y1	y2	y3	...
...	z1	z2	z3	...

Выходные данные. Картинка (необходима для создания мультфильмов или игр) может быть сохранена либо в формате видеопамати, что привязывает ее к конкретному типу адаптера дисплея, либо как список примитивов: набор точек, линий и др., заданных в экранных координатах. Его формат:

...	P1	X1	Y1	X2	Y2	...	Xn	Yn	P2	X1	Y1	...
-----	----	----	----	----	----	-----	----	----	----	----	----	-----

P1, P2 ... — тип примитива.
X1, Y1, X2, Y2 ... — параметры примитива. Их количество зависит от типа примитива (точка — 2 коорд., линия — 4 коорд. и т. д.).

Формулы. Над фигурой проводятся операции: сдвиг, поворот, пересчет в пространство ЛП. Координаты каждой точки в Декартовом пространстве X_n, Y_n и Z_n обрабатываются по приведенным ниже формулам. Для преобразования фигуры, представляющей из себя массив координат, необходимо пересчитать все координаты фигуры, записанные в массиве.

Поворот:

X_n, Y_n, Z_n — начальные координаты точки.

X'_n, Y'_n, Z'_n — промежуточные.

X''_n, Y''_n, Z''_n — конечные координаты.

A — угол поворота вокруг оси OX, B — вокруг OY, C — вокруг OZ.

Поворот по всем трем осям осуществляется последовательно:

вокруг оси OX

$$Y_n = Y_n^* \cos(A) + Z_n^* \sin(A)$$

$$Z_n = Z_n^* \cos(A) - Y_n^* \sin(A)$$

вокруг оси OY

$$X_n = X_n^* \cos(B) + Z_n^* \sin(B)$$

$$Z_n = Z_n^* \cos(B) - X_n^* \sin(B)$$

вокруг оси OZ

$$X_k = X_n^* \cos(C) + Y_n^* \sin(C)$$

$$Y_k = Y_n^* \cos(C) - X_n^* \sin(C)$$

Сдвиг:

dX — сдвиг по оси X, dY — по оси Y, dZ — по оси Z.

$$X_k = X_n + dX$$

$$Y_k = Y_n + dY$$

$$Z_k = Z_n + dZ$$

Формулы ЛП:

X_n, Y_n, Z_n — координаты точки в пространстве.

X_k, Y_k — координаты точки на экране.

X_ϕ, Y_ϕ — координаты центра экрана.

K — масштабный коэффициент.

$$K = \frac{R_2}{R_0 + Z_n}$$

R_2 — расстояние от центра проектирования (глаз) до экрана.

R_0 — расстояние от глаза до точки, условно принимаемой за центр координат пространства ЛП.

$$X_k = X_\phi + K \cdot (X_n - X_\phi) + X_\phi$$

$$Y_k = Y_\phi + K \cdot (Y_n - Y_\phi) + Y_\phi$$

Эти формулы описывают центральное проектирование изображения в пространстве на плоскость (экран):

O — проекция центра координат на ось взгляда.

G — глаз.

A — исходная точка.

X — искомая точка, полученная центральным проектированием точки A на экран.

$$X(X_k, R_2 - R_0)$$

$\triangle ABG$ подобен $\triangle XCG \Rightarrow \Rightarrow CX/AB = CG/BG$

$$\frac{X_k - X_\phi}{X_n - X_\phi} = \frac{R_2}{R_0 + Z_n} \Rightarrow$$

$$\Rightarrow X_k = K \cdot (X_n - X_\phi) + X_\phi, \text{ где}$$

$$K = \frac{R_2}{R_0 + Z_n};$$

X_ϕ добавляется для сдвига изображения к центру экрана. Формулы для X_k выводятся аналогично.

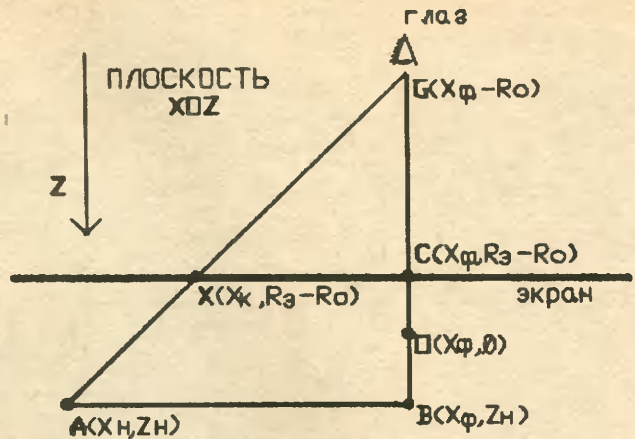
Запишем все формулы в объединенном виде:

$$X_k = X_\phi + \frac{R_2 \cdot ((X_n \cdot \cos(B) + (Z_n \cdot \cos(A) - Y_n \cdot \sin(A)) \cdot \sin(B)) \cdot \cos(C) + (Y_n \cdot \cos(A) + Z_n \cdot \sin(A)) \cdot \sin(C) - X_\phi + dX) - X_n \cdot \sin(B) + dZ}{R_0 + (Z_n \cdot \cos(A) - Y_n \cdot \sin(A)) \cdot \cos(B) - X_n \cdot \sin(B) + dZ} + X_\phi$$

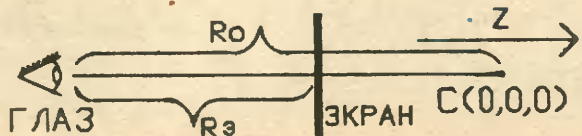
$$Y_k = Y_\phi + \frac{R_2 \cdot ((Y_n \cdot \cos(A) + Z_n \cdot \sin(A)) \cdot \cos(C) - (X_n \cdot \cos(B) + (Z_n \cdot \cos(A) - Y_n \cdot \sin(A)) \cdot \sin(B)) \cdot \sin(C) - Y_\phi + dY) - X_n \cdot \sin(B) + dZ}{R_0 + (Z_n \cdot \cos(A) - Y_n \cdot \sin(A)) \cdot \cos(B) - X_n \cdot \sin(B) + dZ} + Y_\phi$$

X_ϕ, Y_ϕ — координаты точки схода (по «Z» она считается бесконечно удаленной).

В пространстве глаз имеет координаты ($X_\phi, Y_\phi, -R_0$). Экран параллелен плоскости XOY и ее «Z» коорд. равна ($R_0 - R_2$).



В массиве хранятся начальные координаты фигуры, которые после преобразований по формулам ЛП не сохраняются, так как не имеют смысла в Декартовом пространстве (они имеют смысл только на экране). После сдвига координаты записывать также не стоит, так как это приведет к смещению фигуры относительно центра координат и при очередном повороте она будет поворачиваться уже не вокруг своей оси, а вокруг центра координат и одновременно с поворотом будет двигаться по дуге. При этом будет труднее определить место, куда же она в итоге попадет. Только при операции поворота



координаты после пересчета имеет смысл заносить в массив.

Когда требуется сформировать на экране изображение сдвинутой и повернутой в пространстве ЛП фигуры, нужно сначала ее повернуть, а потом уже перемещать и формировать изображение, не сохраняя при этом координаты в массиве.

В целях экономии времени достаточно просчитывать по формулам только упрощенный вид фигуры (вершины и направляющие), т. е. каркасно-проволочную модель, а мелкие детали дорисовывать в графическом редакторе от руки.

Алгоритмы разработаны автором в МКБ (межшкольном конструкторском бюро) ИПИ АН СССР и реализованы на языках BASIC, СИ в виде двух программ. Первая представляет из себя графический редактор, позволяющий моделировать в диалоговом режиме размещение фигур в пространстве ЛП и сохранять последовательность операций преобразования в текстовом файле. Также имеется возможность сохранять картинки изложенными выше способами.

Вторая программа — распаковщик созданных файлов. Она считывает из текстового файла операции и выполняет их, формируя сложное графическое изображение из многих фигур, которое потом можно сохранить. Ниже приведен фрагмент этой программы, реализующий вышеизложенные алгоритмы:

Список переменных:
 Флаги: F1: поворот; F2: сдвиг; F3: перспектива;
 (0 — запретить 1 — разрешить).
 CO — цвет фигуры. DX, DY, DZ — величины сдвигов. A, B, C — углы поворота. RO, RE, XF, YF — параметры про-

```

10 ' НАЧАЛО
20 ' параметры : F1, F2, F3, A, B, C, DX, DY, DZ, RO, RE, XF, YF
30 ' промежуточные константы поворота
40 ' IF F1=0 THEN 70
50 SA=SIN(A); CA=COS(A); SB=SIN(B)
60 SC=SIN(C); CC=COS(C); CB=COS(B)
70 ' Обработка массива с выводом
80 FOR I=0 TO MM: X=M(I, I); Y=M(I, I); Z=M(I, I)
90 ' IF F1=0 THEN 140
100 ' формулы поворота
110 IF UA<0 THEN D=Y: Y=YXCA+ZXSA: Z=ZXCA-DXSA
120 IF UB<0 THEN D=X: X=XXCB+ZXS: Z=ZXCB-DXSB
130 IF UC<0 THEN D=X: X=XXCC+YXSC: Y=YXCC-DXSC
140 ' IF F2=0 THEN 170
150 ' формулы сдвига
160 X=X+DXXR: Y=Y+DYXR: Z=Z+DZXR
170 ' IF F3=0 THEN 210
180 ' формулы перспективы
190 K=RE/(RO+Z/R)
200 X=(KX(X/R-XF)+XF)XR: Y=(KY(Y/R-YF)+YF)YR
210 ' вывод на экран
220 X=INT(XE+X/R+.5): Y=INT(YE-Y/R+.5)
230 IF IMOD2=0 THEN PSET(X, Y), CO ELSE LINE-(X, Y), CO
240 NEXT I: RETURN
  
```

ФУНКЦИЯ ОБРАБОТКИ ФИГУРЫ
 (Для Turbo C++)

Параметры функции :

- flag: turn — флаг поворота,
- scale — масштабирования,
- move — сдвига,
- persp — перспективы.

Указатель на исходную фигуру *inp* и конечную *out*.

Фигура определена как структура:

```

typedef
struct {
  int SConst, /*X масштабный множитель X/
  lenf, /*X номер последнего элемента X/
  arr[100][3]; /*X массив координат X/
} f_type;
  
```

Xp — указатель на массив, содержащий список параметров (углов поворота, коэффициентов масштабирования и пр. в зависимости от флагов).

```

/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/
#include <math.h>

int f_treat( int turn, int scale, int move, int persp,
  f_typeX inp, f_typeX out, double Xp)
{
  int i, p_schet=0;
  double dx, dy, dz, xf, yf, ro, re, x, y, z, d, sa, sb, sc, ca, cb,
  cc, ua, ub, uc, kx, ky, kz, k, coeff=0.017453;
  /X----- чтение массива параметров -----X/
  if (turn==1) {
    ua=coeffXp[0]; ub=coeffXp[1]; uc=coeffXp[2];
    sa=sin(ua); ca=cos(ua); sb=sin(ub);
    cb=cos(ub); sc=sin(uc); cc=cos(uc); p_schet+=3;
  }
  if (scale==1) {
    kx=p[p_schet++]; ky=p[p_schet++]; kz=p[p_schet++];
  }
  if (move==1) {
    dx=p[p_schet++]; dy=p[p_schet++]; dz=p[p_schet++];
  }
  if (persp==1) {
    xf=p[p_schet++]; yf=p[p_schet++];
    ro=p[p_schet++]; re=p[p_schet++];
  }
  /XXXXXXXXXX начало цикла обработки фигуры XXXXXXXXXXXX/
  for (i=0; i<=inp->lenf+b_bii++) {
  /X----- взятие очередной точки -----X/
    x=(double)inp->arr[i][0];
    y=(double)inp->arr[i][1];
    z=(double)inp->arr[i][2];
  
```



```

/X----- поворот -----X/
  if (turn==1) {
    if (ua!=0.0) { d=y; y=y*ca+z*sa; z=z*ca-d*sa; }
    if (ub!=0.0) { d=x; x=x*cb+z*sb; z=z*cb-d*sb; }
    if (uc!=0.0) { d=x; x=x*cc+y*sc; y=y*cc-d*sc; }
  }

/X----- масштабирование -----X/
  if (scale==1) {
    x*=kx; y*=ky; z*=kz; }

/X----- сдвиг -----X/
  if (move==1) {
    x+=dx*inp->SConst;
    y+=dy*inp->SConst;
    z+=dz*inp->SConst; }

/X----- перспектива -----X/
  if (persp==1) {
    k=re/(ro+z/inp->SConst);
    x=(k*(x/inp->SConst -x*f)+f)*inp->SConst ;
    y=(k*(y/inp->SConst -y*f)+f)*inp->SConst ; }

/X----- запись в конечную фигуру -----X/
  out->arr[i][0]=(int) (x+0.5);
  out->arr[i][1]=(int) (y+0.5);
  out->arr[i][2]=(int) (z+0.5);
}
out->SConst=inp->SConst;
out->lenf=inp->lenf;
return 0;
}

```

108

странства. М [] — массив, содержащий фигуру.
 ММ — номер последнего элемента массива.
 R — масштабный множитель.
 XE, YE — координаты центра экрана.
 С вопросами обращаться по тел: 132-19-33.

Литература

Язык СИ для профессионалов / библиотека программиста — «И.В.К. СОФТ», М.: Металлургия, 1991

К. ФЕОКТИСТОВ, XI кл.
 ШГ № 67, Москва

Реализация ЛОГО для КУВТ-86

Несмотря на растущее число компьютерных классов, оснащенных машинами «Ямаха», УКНЦ и IBM-совместимыми компьютерами, в большей части школ еще остаются комплексы обучающих машин КУВТ-86. Крайняя ограниченность памяти и низкое быстродействие компьютеров БК-0010Ш, использующихся в качестве рабочих мест учеников, а также бедность и примитивность программного обеспечения не позволяют полностью использовать все возможности комплекса КУВТ-86.

Возникает необходимость в простом для освоения программном продукте, который должен хотя бы в какой-то мере компенсировать недостатки БК, ознакомить начинающего с современными методами общения с машиной и обучить его какому-либо простейшему языку программирования.

В 1990 г. автором на языке Фокал был написан интерпретатор языка ЛОГО для БК-0010. Создание программы преследовало две цели: во-первых, собственно написание интерпретатора и, во-вторых, проверка возможности создания на Фокале мощного и современного пользовательского интерфейса.

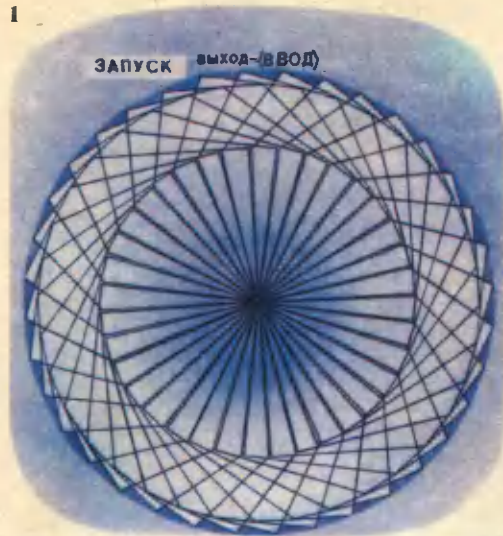
Поддерживаемый интерпретатором русифицированный вариант языка ЛОГО включает в себя основные стандартные операторы, а также специальные процедуры (рисование окружности, многоугольника и др.). Для облегчения набора текстов программ можно сокращать названия операторов до трех первых букв. Предусматривается возможность подключения к интерпретатору библиотек процедур, расширяющих его возможности.

Приводимый ниже пример (рис. 1) соответствует версии 4.0 интерпретатора.

Пример 1. Рисование «подсолнуха».

ЭТО ЧАСТЬ
 ПРОЦЕДУРА ДОМ 50
 ВПРАВО 10
 ВСЕ
 ПОВТОРИТЬ 35: ДЕЛАЙ
 ЧАСТЬ

Для повышения скорости исполнения программ пользователя предусмотрена их частичная компиляция перед запуском на выполнение. В процессе компиляции осуществляется контроль за ошибками в исходном тексте с выдачей соответствующих сообщений. После обнаружения ошибки можно сразу перейти к исправлению неверной строки программы. Для облегчения отладки в интерпретаторе предусмотрен режим пошагового выполнения программы, причем одновременно отображаются исполняемый опера-



тор и результаты, полученные в предыдущих и данном шагах.

Системы меню и «горячие» клавиши облегчают работу с программой, а также знакомят с современными методами реализации пользовательского интерфейса.

Встроенный редактор предоставляет широкий набор возможностей для написания и редактирования текста программы на ЛОГО. Работу с ним облегчает специальное меню, в котором перечислены все команды редактора.

Так как интерпретатор предназначен для начинающего пользователя, в нем реализована система помощи, дающая краткую иллюстрированную справку о его возможностях и языке ЛОГО, причем для каждого оператора приводится пример использования, краткая запись и «картинка», иллюстрирующая выполняемые им действия.

Интерпретатор обеспечивает также возможность загрузки существующих и записи имеющихся файлов с помощью магнитофона на магнитную ленту. Данная программа может применяться на всех комплексах КУВТ-86, где на БК-0010

используется язык Фокал, а также просто на БК-0010 при наличии любого дисплея.

Возможно, в ближайшее время появится и программа связи, которая будет пересылать файл, созданный интерпретатором, с БК на ДВК-2МШ для записи его на магнитный диск, а также загружать файл на БК с магнитного накопителя ДВК.

Далее мне хотелось бы поделиться выводами, полученными при создании интерпретатора.

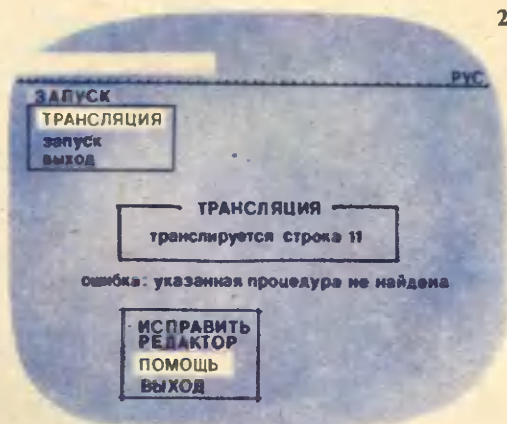
Главным недостатком Фокала является низкое быстродействие, и для его увеличения приходится применять ряд специальных приемов: тщательный анализ алгоритмов для уменьшения времени их исполнения;

вынос из циклов независимых от переменной цикла выражений;

вычисление всех известных арифметических выражений; ликвидация излишних переменных;

применение специальных приемов программирования [2].

Не старайтесь оптимизировать всю программу, обращайте внимание только на критические по времени исполнения участки.



Весьма распространенной задачей в компьютерной графике является рисование окружности. Для этого применяется ряд алгоритмов, но все они выполняются достаточно медленно. В процессе написания интерпретатора автором был разработан алгоритм получения на экране восьми (!) точек окружности в течение одного выполнения тела цикла.

```

1.1 F I=0,R/1.42;
    S Y=FSQD(RXR-XXX);D 2
1.2 D
2.1 X FT(1,X0+X,Y0+Y/1.4);
    X FT(1,X0+X,Y0-Y/1.4);
    X FT(1,X0+Y,Y0+X/1.4)
2.2 X FT(1,X0+Y,Y0-X/1.4);
    X FT(1,X0-X,Y0+Y/1.4);
    X FT(1,X0-X,Y0-Y/1.4)
2.3 X FT(1,X0-Y,Y0+X/1.4);
    X FT(1,X0-Y,Y0-X/1.4)
  
```

В этом примере X_0 и Y_0 — координаты центра окружности, а R — ее радиус. Обратите внимание, что в строке 1.1 значение $\sqrt{2}$ указано явно.

Современный пользовательский интерфейс нельзя представить без систем меню. Ниже приводится пример реализации меню на Фокале. В группе строк 2 нужно указать необходимые координаты и названия пунктов меню. При желании можно заменить оператор в строке 4.2 на оператор чтения регистра данных клавиатуры

Наконец, введите в П7, 1 П9, запас бомб, число пулеметных лент и длину трассы в километрах: 15 П3, 20 П2, 30 П1.

Можно стартовать. Вы вооружены пулемотом и бомбами, враги — зенитками, «Стингерами» и летающими зондами. Начинайте первый полет —

ВП 13 С/П

Первая цифра на индикаторе — высота полета (от 1 до 9 км). Сейчас можно изменить высоту: Р — вниз на 1 км, ГРД — на той же высоте, Г — вверх на 1 км.

Через несколько секунд мелькнет обзор 7-километрового пространства перед «Вертолом». Обозначения здесь такие: крайняя слева восьмерка — «Вертол»; единица — вражеская зенитка или ракета; числа от 2 до 9 — зонды-разведчики, летящие на высоте, соответствующей цифре; 0 — пустое пространство.

«Вертол» летит со скоростью 1 км за маневр. Действуйте: Р — сброс бомбы, ГРД — без огня, Г — пулеметная очередь (одна лента). Для поражения зонда нужно стрелять, находясь с ним на одной высо-

те. При попадании он может исчезнуть, изменить высоту или даже превратиться в зенитку или ракету. Зенитку или ракету можно поразить только бомбой (при этом можно «зацепить» и зонд). Пролететь мимо зонда можно только в том случае, если разница высот не менее 2 км — иначе вы будете уничтожены.

При аварии или поражении ракетой или зениткой на индикаторе высвечивается ЕГГОГ. Если кончились боеприпасы, при попытке выстрела выдается сообщение СЕЛ. После успешного полета — сообщение ВСЕ; следующий полет можно начать с С/П, предварительно заполнив регистры.

Итак, последний слуга Во Ланда пал, сраженный пулеметной очередью. Брюс Ли сажает «Вертол» у неприступных скал, в которых укрыто последнее убежище Во Ланда — «Тайный бункер». Но, чтобы открыть бронированную дверь, необходимо поразить десятью выстрелами шесть мишеней. Мишени не стоят на месте, а движутся на вас или от вас. Моделирует эту ситуацию программа «Тир-2» С. Тарасова и А. Клишина (Москва).

Переключатель Р—Г поставьте в положение Г. Введите

8.555555 П7, <произвольное восьмиразрядное число> П6, 1 ВП 30 Fx² Fx² ПА

Начало игры:

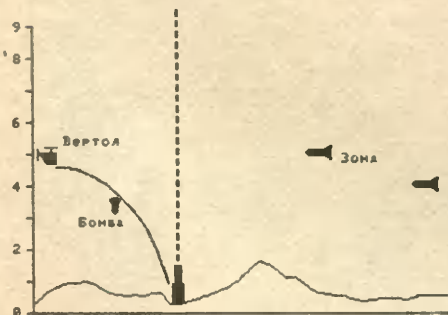
В/0 С/П

Имеется шесть мишеней, каждая шириной 1 м, расстояние между ними 1 м. На экране:

в.-----в

В РУ — расстояние до мишеней. Введите угол прицеливания (отрицательный — влево, положительный — вправо) и стреляйте — С/П. На экране появится положение после выстрела. Когда все мишени сбиты, в РУ — количество затраченных выстрелов. Если десяти выстрелов не хватило, высвечивается IIIГГОГ.

Выполнить поставленную задачу непросто, но можно, и в конце концов перед вами откроется, скрипя, дверь бункера. Что ждет вас там? Об этом — в следующем выпуске. А пока ждем ваших писем с оценкой программ (сервис, игра, общая оценка) по десятибалльной системе. Будем рады всем замечаниям и предложе-



0	1	2	3	4	5	6	7	8	9
0 Сx	П5	ИП7	П8	ИП6	ВТ	КСЧ	П6	2	1
1 X	5	+	К[x]	П9	7	ИП8	ВП	7	+
2 КИВ	С/П	КИП5	Xв	Ftg	ИП9	X	К[x]	2	1
3 К[x]	Fx+0	04	FBx	К[x]	Fx<0	39	1	-	4
4 +	/-/	F10x	К[x]	5	X	8	+	ИП8	Xв
5 КА	К0	П8	К[x]	Fx=0	04	ИП5	ВП	15	

ниям, вашим собственным программам.

До новых встреч!

В. МОСКОВЦЕВ

80 100504

Поправка. По вине редакции в программе «Шашки 12» (статья «Этиод в черно-белых тонах», «ИНФО» 4—91) появились опечатки. По адресам 35—39 должны быть следующие команды: 35./—/ 36.1 37.КППЕ 38.3 39.—

По адресу 60 должно стоять ИПВ.

Компьютеры в обучении: шведский путь

Брошюра под таким названием издана в 1990 г. шведским Национальным советом по образованию совместно с издательским советом «Компьютеры в обучении». В число авторов входят исследователи, педагоги, руководители образования, включая генерального директора Национального совета по образованию Е. Рингборга. Содержание данной брошюры (с некоторыми сокращениями) приводится ниже.

Шведская школьная система

В Швеции дети идут в школу в 7 лет. К этому времени большинство из них по меньшей мере год посещают дошкольные учреждения. Обязательное школьное обучение длится 9 лет, после чего около 90 % учеников переходят в старшую среднюю (*Upper secondary schools*) школу, где получают либо профессиональную, либо дальнейшую общеобразовательную подготовку. Около 25 % из них поступают затем в высшие учебные заведения. Несколько миллионов юношей и девушек старше 18 лет охвачены также различными формами образования взрослых.

Все школьное обучение бесплатное. Оно основано на принципе равенства базового уровня образования (для всех детей, независимо, например, от места жительства или дохода родителей). Именно поэтому его цели и главные направления определяются риксдагом (парламентом). Имеется Национальный совет по образованию, отвечающий за обязательное обучение, старшие средние школы и образование взрослых. Существуют также местные органы управления образованием — 24 окружных комитета, муниципальные комитеты.

Этапы компьютеризации

В 70-х гг. Национальный совет по образованию совместно с университетами и колледжами начал осуществление ряда исследовательских проектов по использованию компьютера как педагогического средства. Одновременно в нескольких старших средних школах было введено обучение программированию в рамках математики и некоторых технических предметов.

Проекты были завершены к началу 80-х гг., и их опыт позволил определить пути использования компьютеров в школах. Основной вывод состоит в том, что компьютер следует рассматривать как один из компонентов целостной системы учебных средств. Считается, что «натаскивание» с помощью компьютера не представляет сколь-нибудь значительной ценности, а знания в области программирования не являются необходимой предпосылкой освоения компьютера. Были сформулированы требования к школьному компьютеру, основанные на перспективах его использования в учебном процессе.

В 1983 г. была принята программа широкомасштабного внедрения компьютеров в школы. По решению парламента изучение



компьютеров (*computer studies*) стало обязательным предметом для учеников 13—16 лет, причем предметом, интегрированным с другими школьными дисциплинами. Школы получили государственные субсидии для приобретения компьютеров, удовлетворяющих требованиям учебного процесса, 1—2 учителя из каждой школы прошли соответствующую подготовку. Началось использование компьютеров в профессиональном образовании.

В 1985 г. Национальный совет по образованию поручил рабочей группе «Педагогические программные средства» определить основные направления разработки указанных средств и обеспечить их внедрение на локальном и региональном уровнях. На основе полученных группой результатов в 1988 г. была начата реализация проекта «Компьютеры и образование», рассчитанного на 3 года. Работа велась по следующим направлениям: 1) пилотажные исследования по широкому кругу проблем, связанных с возможностями и ограничениями использования компьютера как средства обучения, 2) разработка программного обеспечения по отдельным предметам (в основном программ на имитационное моделирование, 3) широкомасштабные проекты на национальном и региональном уровнях, включая создание учебных средств с использованием постоянной памяти на компакт-дисках и интерактивных видеосистем, а также применение компьютеров для обучения математике и музыке.

Начиная с 1983 г. Национальный совет по образованию ежегодно проводит оценку школьных компьютеров, причем критерии оценки постоянно повышаются. Так, в 1986 г. было введено требование учета особенностей обучения детей с отклонениями. Особое внимание уделяется также возможности объединения школьных компьютеров в локальную сеть. Сегодня (к 1990 г., прим. ред.) в школах Швеции установлено не менее 27 000 персональных компьютеров, т. е. 1 компьютер на 40 учеников.

При оценивании компьютеров проводится также апробация предназначенных для них средств широкого назначения — текстовых редакторов, электронных таблиц и т. д. Ряд исследовательских проектов посвящен анализу педагогических преимуществ использования этих средств в учебном процессе как альтернативного пути, разработке специального программного обеспечения для школ.

Десятилетний опыт компьютеризации школ Швеции показал, что компьютер обладает значительным потенциалом. В будущем необходимо реализовать этот потен-

циал как можно более полно. Для этого, в частности, учителям необходимо обучать не только общим принципам работы с компьютером, но и использованию их при обучении конкретным школьным предметам. Важной задачей остается также подготовка учащихся к активной, полноценной жизни и работе в условиях технологически развитого общества.

Применение компьютеров в школе Тенста

Наглядным примером применения компьютеров является обучение в школе Тенста. В этой старшей средней школе компьютеры используются в рамках:

- а) обучения работе на станках с числовым программным управлением;
- б) курсов по экономике, торговле и управлению (текстовые редакторы, настольные издательства, базы данных, средства ведения записей);
- в) практикума по физике;
- г) доступа к различным базам данных (как правило, с помощью школьного библиотекаря).

Учениками школы полностью подготовлена к изданию книга по теме «Язык, живопись и компьютеры». Книга профессионального качества распространяется коммерчески, причем работа по маркетингу выполнена учениками, изучающими экономику.

Компьютеры и процесс письма

Развитие навыков чтения и письма является важным компонентом школьного обучения. Как показывает опыт, значительную роль в этом могут сыграть компьютеры. Уже в младших и средних классах школьники (хотя и не все) могут успешно осваивать и использовать компьютер как орудие письма, что приводит к развитию творческого мышления и повышению интереса учащихся.

При обучении старших школьников компьютер использовался на всех этапах подготовки текста — как предшествующих собственно письму (анализ, сбор и сортировка материала, предварительное структурирование и планирование текста), так и следующих за ним (модификация, корректировка, распечатка). Компьютер позволяет сосредоточиться на содержании, оставив на потом проверку орфографии или деталей изложения. Распечатка промежуточных версий помогает яснее увидеть и при необходимости изменить его структуру. Кроме того, при работе с компьютером более наглядно выступают формальные требования к тексту и некоторые особенности языка.

Новые возможности, которые представляют компьютеры, — математические практикумы (буквально — «математические мастерские») и системы манипулирования символами, — имеют для обучения математике большое значение. Для исследования проблем, связанных с изменением характера и методов обучения математике в условиях применения ЭВМ, была начата реализация проекта ADM (аббревиатура шведского названия «Анализ роли компьютеров в преподавании математики»).

Традиционно при обучении математике основной акцент делается на непосредственном оперировании с матриалами — вычислениях, решении уравнений, поиске максимальных и минимальных значений и т. д. Сейчас эти операции можно выполнять с помощью компьютера и появились условия для того, чтобы обратить больше внимания на успешность понимания задачи, планирования решения, оценки полученных результатов.

Первые шаги в этом направлении в рамках проекта ADM были предприняты в 1987 г. При обучении математике в XI—XII классах (возраст учащихся — 17—19 лет) использовались электронные таблицы, система компьютерной алгебры и «Математическая мастерская». В дальнейших исследованиях применялась только последняя из этих систем. Анализировались следующие вопросы: каким должен быть курс методов вычислений в условиях, когда почти все рутинные действия выполняются с помощью «Математической мастерской», какие требования предъявляют к этой системе школьники и учителя, приводит ли её применение к повышению уровня знаний и способностей учащихся.

Опыт экспериментального обучения математическому анализу и дифференциальным уравнениям (включая использование последних для описания ситуаций) показал, что применение «Математической мастерской» стимулирует активную работу учащихся и обсуждение ими математических идей. Кроме того, расширение круга практических приложений способствует более глубокому пониманию основных понятий. При этом успешность обучения в первую очередь зависит от квалификации и установок учителя.

Таким образом, применение «Математической мастерской» обещает повысить общий уровень преподавания математики в школе. Для этого, однако, необходима большая работа по перестройке традиционной системы обучения.

В Швеции инженерная подготовка начинается в старшей средней школе, где учащимся дают элементарные знания по техническому черчению, сопротивлению материалов и другим базовым инженерным дисциплинам. После специализации, на третьем году обучения, они получают углубленную подготовку в выбранном направлении. Применение компьютерных средств в обучении будущих инженеров является весьма перспективным, тем более что по расписанию они проводят в компьютерном классе довольно много времени (на первом году технического обучения — 5 ч в неделю). Первые результаты использования в обучении САПР «AutoCAD» (эта система широко применяется в промышленности стран Северной Европы, и именно поэтому выбор пал на нее) являются обнадеживающими.

Учащиеся осваивают систему примерно за 10 уроков. После этого они уверенно применяют наиболее часто используемые команды системы и начинают понимать принципы автоматизированного проектирования. Качество полученных чертежей гораздо выше, чем при обычном черчении. Важным достоинством применения САПР является также возможность добавлять или удалять детали изображения.

В результате на рутинной работе экономится время, которое ученики могут посвятить более важному материалу. Пробуя разные методы решения задачи, школьники учатся смотреть на этот процесс с разных точек зрения, подходят к нему творчески и проявляют больший интерес. Меняется роль учителя — после вводного курса он становится скорее лидером группы: обеспечивает условия работы, указывает общее направление, поощряет учеников и делится с ними своими знаниями.

Компьютеры и художественное творчество

Обучение школьников художественному творчеству рассматривается в Швеции как составная часть формирования индивидуальности человека. Графические возможности современных компьютеров не только позволяют сделать более простым обучение традиционным аспектам художественного творчества — композиции и цвету, но и открывают новые перспективы. Компьютерная графика не может, разумеется, заменить живопись, но она обладает собственной эстетикой.

Опыт применения компьютерных средств

на занятиях по художественному творчеству показал, что эти средства создают условия для экспериментирования с различными вариантами изображения и для синтеза различных видов визуальной информации (например, графики и текста).

Новые возможности для детей с нарушениями развития

В последние годы в Швеции компьютеры все шире используются в школах для детей с различными отклонениями в развитии. Решение начать исследование в этом направлении было принято парламентом и правительством страны. Парламент ежегодно выделяет на эти цели значительные средства. Кроме того, финансируется разработка специального аппаратного обеспечения, а в 1988 г. правительство выделило средства на трехлетнюю программу экспериментальных исследований использования компьютерных средств для помощи детям с нарушениями, которой охвачено 6 регионов. Кроме того, финансирование осуществляется самими региональными центрами, а также Шведским институтом дефектологии и Шведским институтом учебных материалов (разрабатывает специальные педагогические средства для учащихся с физическими нарушениями слуха и зрения, задержками умственного развития).

Применение информационных технологий, использование компьютеров со специальными программным обеспечением и периферией открывают новые возможности для обучения детей с различными отклонениями. Однако для полной реализации необходимо решить целый ряд проблем, связанных с созданием методик и программ, их оценкой, распространением и поддержкой на национальном уровне. Упомянувшиеся выше институты с 1985 г. осуществляют совместный проект, направленный на решение указанных проблем.

Изучение второго языка

Значительная часть программного обеспечения, предназначенного для обучения иностранному языку, основана, по существу, на принципах программированного обучения, популярного в 1960-х гг. Однако в последние годы акцент делается на создании учебных сред, которые поощряют собственную активность учащихся, освоение коммуникативных функций языка, поиск ответов на самостоятельно поставленные вопросы.

В экспериментальных исследованиях показана эффективность использования систем текстообработки, систем проверки орфографии, компьютерных тезаурусов, гипертекстов

и баз данных (в частности, обмен данными с зарубежными базами) для поддержки творческой активности учащихся, направленной на создание, преобразование и анализ текстов.

Средства поддержки по социальным дисциплинам

Необходимость быть в курсе текущих событий особенно важна для учителей по социальным дисциплинам, так как каждый год появляются все новые и новые важные материалы. Одно из возможных решений возникающих проблем состоит в использовании баз данных.

Экспериментальное обучение социальным дисциплинам с помощью компьютеров проводилось в рамках проекта *ADS*, участвовали 10 старших средних и 4 базовые школы. Оценка программного обеспечения, специально предназначенного для обучения социальным дисциплинам, не обнаружила его существенного влияния на качество обучения. Поэтому в настоящее время основное внимание уделяется использованию компьютера в качестве такого средства, каким он является в обычных исследованиях по социальным наукам.

В рамках проекта разработаны компакт-диски с информацией о Швеции. По мнению учителей, распространение новых версий такого рода баз данных 1—2 раза в год является вполне достаточным, чтобы избежать их устаревания.

Обучение музыке с помощью компьютеров

Компьютеры позволяют почувствовать вкус музыкального творчества даже тем учащимся, которые не владеют ни одним музыкальным инструментом. Кроме того, интерес к компьютерам приводит к повышению интереса и к занятиям музыкой (особенно у мальчиков).

Имеющийся опыт показывает, что существующее программное обеспечение имеет как достоинства, так и недостатки. С одной стороны, оно обеспечивает свободу творчества, но с другой — не отвечает комплексным требованиям процесса обучения. Чтобы охватить все этапы этого процесса, необходимо использовать разные версии не только программного, но и аппаратного обеспечения, что крайне неудобно и дорого.

Другая проблема состоит в том, что чем мощнее средство, тем больше времени на освоение оно требует. Поэтому педагогические институты должны как можно скорее включить курсы компьютерной грамотности в свои учебные программы.

Сотрудничество стран Северной Европы

Сотрудничество стран Северной Европы осуществлялось Комитетом по педагогическим программным средствам и технологиям, назначенным Советом Министров Северных стран. Основные направления: распространение информации, стандартизация компьютерных средств, разработка технологий и программных продуктов. В частности, разработаны средства проектирования *Mosaikk*, моделирования *SimTek* и переноса программ на разные типы машин с разной графикой *GrafTrix*.

В разработке педагогических программных средств активную роль играли учителя. Они предлагали идеи и затем участвовали в процессе разработки. Организован ряд семинаров и школ по оценке различных типов систем и по подготовке учителей. В рамках проекта *Nordic 90* разрабатывались интерактивные видеосистемы, база данных *Nordic Database*, справочные материалы на *CD ROM*, учебная база данных для людей с нарушениями слуха.

Учителя

Компьютеризация школы не дает результатов, если ее не поддерживают учителя. В 1984 г. была организована Шведская ассоциация учителей, использующих компьютеры, которая ежегодно в сотрудничестве с Национальным советом по образованию организует национальные конференции.

Одним из самых острых вопросов является ответственность за компьютеры. Обычно ответственными (как в педагогическом, так и в техническом плане) являются один или несколько учителей, которым в виде компенсации снижается учебная нагрузка (как правило, примерно на 3 ч в неделю). В некоторых старших средних школах техническое обслуживание обеспечивается отдельно.

Последние 4 года ассоциация выпускает ежеквартальный журнал «Компьютеры в обучении».

Профессиональная подготовка в современном обществе

Развитие информационных технологий представляет собой один из основных факторов, влияющих на изменение рынка труда. Цель профессионального обучения в этих условиях состоит не в том, чтобы заменить старые знания новыми, а скорее в дополнении первых, так как современные компьютерные системы создаются с учетом потребностей и умений «некомпьютерных» профессионалов.

Школьное обучение должно обеспечивать

понимание принципов работы разнообразных компьютерных средств, умений работы с разными типами аппаратного и программного обеспечения. Для этого учащиеся должны иметь доступ к оборудованию и программам, применяющимся в реальной профессиональной деятельности.

В некоторых случаях ответственность за профессиональное обучение целесообразно перекладывать на заинтересованные фирмы и учреждения, а в школах обеспечивать работу учащихся лишь с компьютерными моделями соответствующих систем. Следует учитывать также, что содержание некоторых видов профессиональной деятельности в условиях компьютеризации претерпевает изменения.

Актуальным является вопрос о том, следует ли учить «ручному» исполнению действий, которые затем автоматизируются с помощью компьютерных средств (черчению, бухгалтерским расчетам и т. д.).

Опыт экспериментального применения систем автоматизированного проектирования в техническом обучении и бухгалтерского учета в курсах по бизнесу показал, что компьютеризация профессионального обучения повышает активность учащихся, позволяет сэкономить время на рутинных действиях и сосредоточиться на содержании решаемых задач.

В некоторых школах проводятся эксперименты с системами, основанными на знаниях. Показано, что необходимость структурировать свои знания в виде формального набора правил значительно повышает уровень понимания материала, даже если полученная в результате система оказывается малоэффективной.

Наконец, разработаны моделирующие программы, позволяющие ученикам познакомиться с технологическими процессами (например, с дистилляцией различных жидкостей), действием лекарственных препаратов, влиянием радиоактивности, имитировать работу оператора атомной электростанции.

Вступая в информационное общество

Информатизация общества требует формирования навыков работы с данными, позволяющих не утонуть в информационных потоках.

В школах Швеции используются различные типы баз данных (БД):

на компакт-дисках (несколько таких БД создано в рамках проекта *CD Project* в педагогическом институте г. Уппсала);

на дискетах и жестких дисках (эти БД распространяются на дискетах, а в школах помещаются на жесткие диски; имеются БД такого рода по современной Швеции, по

профориентации и т. д.); доступ к удаленным БД по телефонным линиям связи;

локальные БД, создаваемые самими учащимися.

Наиболее эффективной формой обучения работе с массами информации представляется выполнение учащимися собственных проектов, требующее сбора, сопоставления, анализа информации, постановки вопросов, ответы на которые можно получить с помощью компьютера. Полезными в этой связи представляются пакеты статистического анализа и графического представления данных. Другим перспективным направлением является использование электронной почты, что подтверждает первый опыт ее применения (организация электронных конференций учителей с помощью системы *SkolKOM*, обмен информацией с помощью системы *Videotex*).

Искусственный интеллект в обучении

Использование элементов искусственного интеллекта позволяет максимально полно учесть потребности обучаемого. В интеллектуальных обучающих системах компьютер строит модель знаний учащегося и на основе ее сопоставления с «эталонной» моделью организует дальнейшее обучение. Пока системы такого рода находятся в экспериментальной стадии, они очень мало используются непосредственно в учебном процессе.

Другим направлением являются интеллек-

туальные системы помощи, которые анализируют действия пользователя и при необходимости вмешиваются в его деятельность, помогая ему эффективно использовать возможности системы.

Системы, основанные на знаниях, могут широко применяться в качестве дополнения к традиционным методам обучения. В диалоге с учащимися система может приводить обоснование своих выводов. Система правил, лежащая в основе, может быть легко дополнена новыми правилами.

Объединение двух подходов — «мультимедиа» (компьютерное представление информации в виде звука и графики) и «гипертекст» (многомерные и многоуровневые связи между элементами текста) — получило название «гипермедиа». Перспективы разработки такого рода учебных средств связаны как с их собственной «интеллектуализацией», так и с использованием в качестве дополнения к системам, основанным на знаниях.

Пролог является языком пятого поколения. Этот мощный и одновременно легкий в использовании язык, близкий к языку логики и математики, проходит в школах Швеции проверку как инструмент обучения математике в старшей средней школе и как инструмент подготовки учителей математики для младших классов средней школы.

Перевод подготовлен
В. Н. КАПТЕЛИНЫМ

ЧТО?
МОЖЕТ
ЭВМ

О СОВМЕСТИМОСТИ

Слова о совместимости с IBM PC часто встречаются в рекламных объявлениях продавцов компьютеров и программ. Что же это такое?

Программы, выполняющиеся на фирменном IBM PC, могут, во-первых, использовать функции DOS; во-вторых (это почти обязательно), функции BIOS (базовая система ввода-вывода, Basic input/output system, состоит из подпрограмм обмена с периферийными устройствами, записанных в ПЗУ); в-третьих, опять же BIOS, но не стандартным образом, а обращаясь к каким-то

фрагментам ее подпрограмм; в-четвертых, возможны обращения непосредственно к компонентам схемы компьютера (например, для ускорения вывода графики многие программы обращаются непосредственно к видеопамяти, минуя BIOS). Каждая программа использует ОЗУ, многие — дисководы (в частности, на дисках хранятся подгружаемые модули больших программ).

Таким образом, вполне совместимым с IBM PC оказывается только PC производства IBM или изготовленный с использова-

нием купленных у IBM «ключевых» микросхем. Практически полностью совместимыми являются компьютеры, копирующие архитектуру фирменных и имеющие BIOS фирмы Phoenix. Применительно ко многим программам оказываются совместимыми ПЭВМ с разнообразными «самопальными» имитациями интеловских процессоров и ПЗУ BIOS (в частности, советские), имеющие как минимум 640 Кбайт ОЗУ и дисководы.

Ну а компьютеры без дисководов и со 128—256 Кбайт ОЗУ совместимы с IBM PC в основном по компьютерным играм.



Школьные компьютерные классы, полностью соответствующие требованиям компьютеризации народного образования

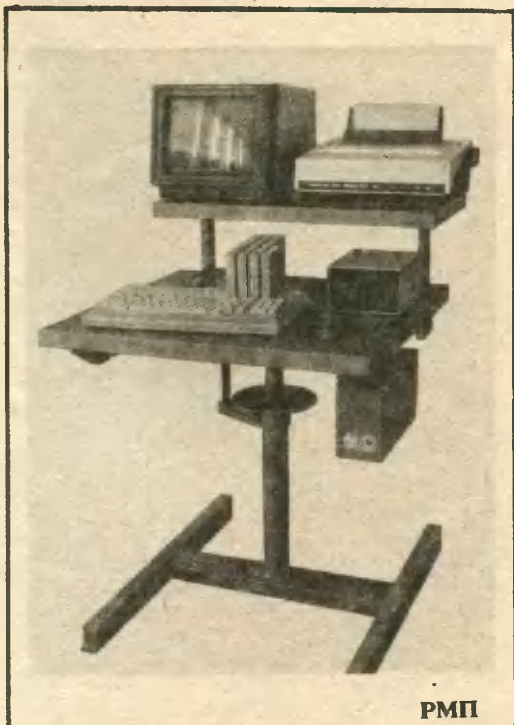
Киевское производственное объединение
«Электронмаш» предлагает

для компьютеризации учебного процесса в школах, профессионально-технических училищах, техникумах, вузах и других учебных заведениях учебно-игровой комплекс (УИК) «Поиск».

УИК «Поиск» — это совместная разработка инженеров, педагогов и врачей, в которой учтены все требования, предъявляемые к школьным компьютерным классам.

УИК «Поиск» — это программная совместимость с компьютерами IBM PC, что соответствует концепции компьютеризации народного образования СССР и позволяет использовать огромные существующие наработки программного обеспечения, в том числе получаемые в рамках реализации проекта «Пилотные школы».

УИК «Поиск» — это безопасность для здоровья детей благодаря размещению компьютеров на специально разработанных в соответствии с санитарно-гигиеническими требованиями и поставляемых потребителю в составе комплекса регулируемых под рост ребенка столах и разработанным врачами рекомендациям по использованию комплексов.



РМП



УНК «Поиск» — это поставка «под ключ» и высококачественный сервис на всей территории Советского Союза.

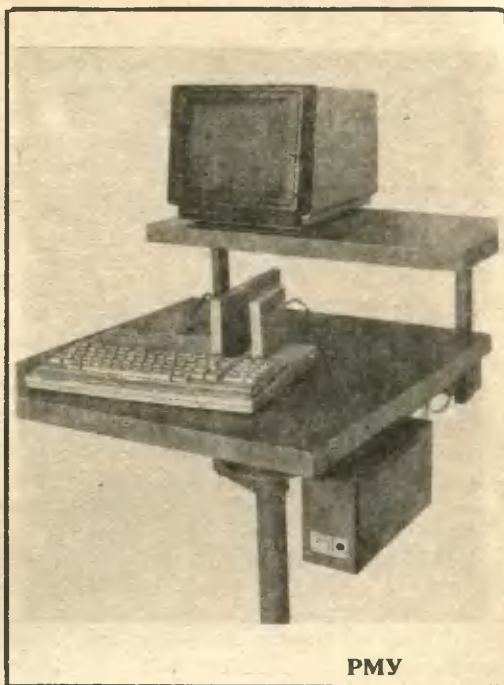
Комплекс «Поиск» представляет собой рабочее место преподавателя (РМП) и от 8 до 15 рабочих мест учащихся (РМУ), объединенных локальной вычислительной сетью со скоростью обмена 64, 300 или 800К бит/с.

По желанию заказчика возможна поставка РМП с ОЗУ 640 Кбайт и внешней памятью «Винчестер» (20—40 Мбайт) и РМУ с монохромным монитором.

Системное программное обеспечение: операционная система типа MS DOS, языки программирования Бейсик, Турбо-Паскаль.

Прикладное программное обеспечение: курс информатики и вычислительной техники, разделы по математике, физике, биологии, украинскому языку и другим предметам.

По желанию заказчика пакеты прикладных программ могут поставляться на русском или украинском языках.



РМУ

Технические характеристики РМП и РМУ

Наименование характеристики	РМП	РМУ
Тип микропроцессора	K1810BM88	K1810BM88
Быстродействие, операций в с	1,25 млн	1,25 млн
Разрядность обрабатываемых данных	8; 16 бит	8; 16 бит
Емкость ОЗУ	360 Кбит	360 Кбайт
Емкость внешней памяти на гибких магнитных дисках	2×1 Мбайт	
Монитор	цветной, разрешение 320×200 точек	цветной, разрешение 320×200 точек
Печатающее устройство	матричное, скорость печати до 100 сим/с	
Стол	регулируемый по высоте	регулируемый по высоте

Стоимость: РМП — 11 400 рублей, РМУ — 6440 рублей (монитор МС6106.02), адаптер локальной сети — 197 рублей.

Для заключения договора на поставку компьютерного класса необходимо направить в наш адрес гарантийное письмо с указанием требуемого количества рабочих мест и платежных реквизитов.

Заявки направлять по адресу:
252180, Киев-180, ул. Б. Окружная, 4.
ПО «Электронмаш».

Телефоны для справок:
отдел маркетинга 475-43-00, 475-98-18
отдел сбыта 475-92-24, 474-12-37
отдел рекламы 474-12-34

Учитель — менеджер

Современная жизнь немыслима без современных информационных технических средств, которые быстро внедряются во все сферы деятельности человека. Введение новых информационных технологий в образование ставит еще одну острую проблему принципиального значения.

120 Хозрасчет и самофинансирование предприятий и организаций, децентрализация народнохозяйственного механизма, выход на самостоятельное развитие регионов, свободный рынок в значительной степени изменяют сложившуюся систему народного образования. В настоящее время рынок представляет пеструю гамму современных технических информационных средств: компьютеров, видеотехники и т. п., а также программно-методических, аудио- и видеоматериалов.

В качестве источников технических и методико-педагогических средств выступают как государственные, так и кооперативные организации. Молодежные центры, различные НТТМ, кооперативы, малые предприятия представляют неоднородную систему по предствительству и уровню квалификации, отношению к сфере образования. Многие из них основывают свою деятельность исключительно на коммерческом подходе к делу, отдавая вопросам бизнеса предпочтение перед научно-методическими проблемами информатизации образования. Нынешняя система народного образования представляет для некоторых дельцов «золотой Клондайк», особенно в периферийных регионах, таких, как Красноярский край, в вопросах поставки компьютеров, других технических средств, их программного и методического обеспечения.

Вполне понятно, что решение проблемы информатизации образования вызывает определенные трудности у педагогов и руководителей различных рангов. Современному руководителю школы или работнику органов образования необходимы знания по основам современных информационных технологий, современному маркетингу и элементарным вопросам менеджмента.

Частичное решение проблемы — переподготовка руководящих кадров. На базе Красноярского педагогического института функционирует факультет повышения педагогических кадров организаторов народного образования (ФППК ОНО). В настоящее время группа директоров школ изучает многоцелевой курс «Менеджер и образование».

Значительную роль в решении обозначенной проблемы может сыграть выпускник вуза, подготовленный как современный организатор, руководитель образования. Лишенный стереотипов, изучающий и занимающийся вопросами современного маркетинга и менеджмента в студенческие годы, получивший педагогическое образование, подобный специалист может эффективно решать вопросы информатизации на местах.

Целесообразность подготовки современного руководителя в среде студенческой молодежи у всех не вызывает сомнений. Однако на реализацию этой задачи имеются различные точки зрения. Действительно, дело новое, серьезное, аналогов в стране нет.

Руководство главного управления народного образования края и ректорат КГПИ решили начать с небольшого. Вышли с предложением в Министерство народного образования РСФСР об открытии в институте новой специализации «Организатор НИТО (новых информационных технологий в образовании)». Вопрос решен положительно. Теперь определенный контингент выпускников института получит две специальности: «учитель по профилю» и «организатор НИТО», что даст возможность молодому специалисту работать в области информатизации образования либо в творческих научных коллективах, например, в роли программистов, сценаристов или режиссеров учебных видеофильмов, либо в должности заместителя директора (завуча), а возможно, и директора школы или работника органов народного образования.

На факультете математики и информатики КГПИ в настоящее время сложились

условия для подготовки подобных специалистов. Полное взаимопонимание и взаимодействие деканата и кафедр, благоприятный морально-психологический климат в коллективе, в котором удачно сочетаются мудрый педагогический опыт старшего поколения и университетский максимализм новосибирской и красноярской школ молодых специалистов, определил по всей видимости тот факт, что КГПИ стал региональным центром информатизации образования.

По оснащенности современными техническими средствами: персональными компьютерами и видеотехникой факультет занимает одно из ведущих мест среди вузов, и не только в крае.

Приоритетное внимание ректората, профсоюза, служб КГПИ к проблемам информатизации, живая связь с Красноярским университетом и другими вузами города и страны позволяют эффективно решать задачи учебно-воспитательного процесса и научных исследований. Ведется долгосрочная хозяйственная работа с ГУНО, большая доля которой приходится на студенческий коллектив.

На факультете действует научно-производственная лаборатория информатики по ремонтно-профилактическому обслуживанию видеокomпьютерных систем школ края, по разработке программно-методических материалов и учебных видеофильмов.

Обучение по новой специализации начинается на старших курсах. Отбор в специализированные группы проводится по итогам учебной и научной работы студентов на младших базовых курсах.

Математический факультет КГПИ ждет заинтересованных абитуриентов. Мы с удовольствием поделимся опытом с заинтересованными вузами и организациями. Ведь речь идет о начальном этапе зарождения новой индустрии образовательной сферы — подготовки и воспитания современного организатора и управленца.

Адрес для справок: 660049, Красноярск, ул. Лебедевой, 89, КГПИ, кафедра информатики. Тел.: 27-12-65.

Н. ПАК, зав. кафедрой информатики КГПИ

121

Автоматизированный учебный курс «Электрик» для КУВТ «Ямаха MSX-2»

Использование компьютера в производственном обучении как средства моделирования целостной профессиональной деятельности позволяет реализовать одно из направлений в технологии широкопрофильной подготовки рабочих электротехнических профессий, объединяющих 24 специальности групп электромонтажников, электромонтеров, электрослесарей.

Работа с компьютером подготавливает учащихся к деятельности в мастерской, формируя профессиональные умения и навыки по:

планированию предстоящего производственного процесса;

чтению принципиальных и монтажных схем;

сборке электротехнических изделий; наладке и пуску собранных изделий; проведению электрических измерений; диагностике определения неисправностей; определению профессиональной квалификации.

Для формирования образа производственного процесса и объекта, умения ориенти-

роваться в трудовых операциях разработан пакет проблемно ориентированных программ имитационного моделирования, автоматизированного учебного курса (АУК) «Электрик» для КУВТ «Ямаха MSX-2».

Структура дидактического сценария АУК «Электрик» в соответствии с учебными целями содержит четыре модели: предметной области, рабочего, обучаемого, учебного процесса.

Основой АУК «Электрик» являются 20 комплексных учебно-производственных упражнений. Обучение идет по уровням производственной деятельности: технология, монтаж, наладка, обслуживание электрооборудования, электроустановок.

Упражнения АУК «Электрик» включают следующие обучающие задания:

планирование предстоящей производственной деятельности;

монтаж элементов электроустановок; соединение элементов в электрическую цепь;

наладка собранных схем; проведение электрических измерений.

Учебная информация модели предмета, находящаяся в памяти ПЭВМ, включает: 43 условно-графических обозначения электрических связей проводами, кабелями и шинами;

23 принципиальных схемы осветительной и силовой сети электрооборудования;

20 видов технологических процессов, включающих от 8 до 18 операций;

трехмерное изображение учебного оборудования: стенды, кабины.

Общение учащегося с компьютером ведется в развернутом диалоге, с использованием режимов «меню» и «подсказка».

Все, кого заинтересовала система АУК «Электрик», могут обращаться за справками по адресу: 443010, г. Самара, ул. Галактионовская, 141. Политехнический институт, ОНИЛ-16, Макаруну Ефиму Абрамовичу.

Е. МАКАРОН, Б. ПРЕСМАН,
С. ТКАЧЕНКО

Делим с любой точностью

Говоря на уроке о бесконечных возможностях ЭВМ, мы, учителя информатики, конечно, лукавим, зная, что наши школьные компьютеры (у кого они есть) далеки от совершенства. И однако же... Все ли их возможности мы исчерпали?

Я смею, например, утверждать, что «Корвет» может точно выдать любое (!!!) число знаков после запятой при делении. Подтверждение тому — программа.

```
10 CLS: PCLS
20 DIM D(1000)
30 INPUT "ВВЕДИТЕ ДЕЛИМОЕ
      И ДЕЛИТЕЛЬ"; A, B
40 S=SGN(A*B): A=ABS(A): B=ABS(B)
50 IF A<>INT(A) THEN A=A*10: B=B*10:
      GOTO 50
60 IF B<>INT(B) THEN A=A*10: B=B*10:
      GOTO 60
70 INPUT "ЧИСЛО ЗНАКОВ
      ПОСЛЕ ЗАПЯТОЙ"; N
80 C=INT(A/B): E=C
90 FOR I=1 TO N
100 D(I)=INT((A-B*C)*10/B)
110 IF A=B*C THEN N=I: GOTO 140
120 A=(A-B*C)*10: E=INT(A/B)
130 NEXT I
140 PRINT "РЕЗУЛЬТАТ"
155 IF S<0 THEN PRINT "-";
160 PRINT C ".";
170 FOR I=1 TO N
180 PRINT D(I);
190 NEXT I
200 END
```

Урок по этой теме я бы начал небольшим лирическим вступлением. Скажем, так: «Из-

вестно, что Фарадей сделал свои первые открытия, имея в распоряжении куски обыкновенной проволоки и обрезки жести. Эдисон сумел восстановить тягу в заводской трубе с помощью мела и веревки. Наверняка и сказочный Левша пользовался не слишком сложным инструментом, хотя для определения размера подковы своей знаменитой блохи ему пришлось снимать мерку с большой точностью.

Большая точность бывает нужна часто, особенно в радиоэлектронике и в машиностроении. И вот вам задача: разделить число А на число В, получив любое наперед заданное количество знаков после запятой. Мы знаем, что для любой ЭВМ число знаков при делении ограничено. Но попробуем снять это ограничение для нашего «Корвета». Попробуем помочь Левше...» (или что-нибудь в этом роде).

Алгоритм решения этой задачи очень прост (он повторяет деление «в столбик») и достаточно ясен из текста программы. Ее можно упростить, сократив или объединив некоторые строки и переменные. Однако в настоящем виде она более наглядна, что очень важно для средней школы. К тому же она позволяет хорошо проиллюстрировать не только возможности ЭВМ, но и многие узловые моменты школьной программы по информатике (циклы, ветвления, массивы).

Е. АЛЕХИН
Брянск

Анкета социолога

На страницах журнала «Информатика и образование», на научных и научно-практических конференциях и семинарах не утихают споры о проблемах информатизации образования. Они касаются не только содержания учебного предмета «Основы информатики и вычислительной техники», его структуры и приоритетов, но затрагивают и более широкий круг вопросов. Таких, например: не становится ли информатика ведущей общеобразовательной дисциплиной в преддверии информационного общества? Действительно ли использование в учебном процессе ЭВТ и НИТ способствует улучшению качества образования? И т. д.

Ни на конкретно-методические, ни на общие вопросы однозначных ответов пока нет. Видимо, они могут быть найдены только на путях серьезного реформирования всей системы образования, что само по себе является серьезной социальной проблемой. Ясно по крайней мере одно: поскольку речь идет о глубинных трансформациях в системе образования, его новых институциональных формах, содержании и целях, постольку необходим комплексный анализ социально-экономических и социально-политических проблем информатизации образования и возможных социокультурных последствий этого процесса.

Однако подобный анализ немыслим без обоснованных научных данных о социальных механизмах и движущих силах столь серьезных изменений. Такие данные может представить социологическая наука.

Начиная это исследование в области социальных проблем использования новых информационных технологий в массовом среднем образовании, кафедра социологии культуры, образования и воспитания МГУ им. М. В. Ломоносова и журнал «Информатика и образование» ставят перед собой целью осмысление реального состояния и перспектив этого процесса. Поскольку систематических и полных данных, характеризующих динамику столь стремительно развивающегося объективного процесса, как использование компьютеров в средней школе, просто нет, мы обращаемся к преподавателям информатики в средних учебных заведениях (школы, гимназии, лицеи, ПТУ и др.), к учителям, использующим в обучении микропроцессорную технику и соответствующие педагогические технологии, к администрации школ с просьбой ответить на ряд вопросов публикуемой анкеты.

123

АНКЕТА

Внимательно прочитайте вопрос и отметьте один или несколько подходящих вариантов ответов, если не указано другое.*

1. Пожалуйста, укажите некоторые сведения о себе:

- вы являетесь преподавателем информатики
- вы преподаете информатику и другой учебный предмет
- вы используете микропроцессорную технику в преподавании традиционных учебных дисциплин
- вы методист или администратор
- другое (что именно?) _____

2. Вы работаете в

- средней общеобразовательной школе
- гимназии или лицее
- профессионально-техническом училище
- колледже
- неполной средней школе
- малокомплектной школе
- в межшкольном УПК
- другое (что именно?) _____

* Для краткости все типы СУЗ в анкете называются, как правило, школами. Если нет специальных указаний, под термином «информатика» подразумевается учебная дисциплина «Основы информатики и вычислительной техники».

3. Ваше учебное заведение расположено

- в крупном городе (население свыше 300 тыс. человек)
- в небольшом городе или поселке городского типа
- в сельской местности
- другое _____

A. Каково реальное состояние школьной информатики?

4. Насколько доступна ЭВТ для учащихся вашей школы?

- компьютерная техника есть в школе
- пользуемся межшкольным кабинетом информатики
- ЭВТ недоступна
- другое (укажите) _____

5. Какой тип ЭВТ вы используете в своей работе? Сколькими машинами вы можете располагать на занятиях одновременно?

	до пяти машин	более пяти машин	машины соединены в сеть
БК-0010	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
«Агат»	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
«Корвет»	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ДВК-2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
УКНЦ	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
«Ямаха»	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
IBM-совместимые	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Микрокалькуляторы	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Разнотипная техника	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Другая техника (какая?)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. Кто обеспечивает вашу школу ЭВТ?

- органы народного образования
- инициативный проект, эксперимент
- шефские предприятия
- собственные средства школы
- другие источники (какие?) _____

7. Как используется ЭВТ в вашей школе?

	не используется	используется регулярно	используется эпизодически
на занятиях по обязательному курсу ОИВТ	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
на занятиях по предметам ЕНЦ	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
на занятиях по предметам ГЦ	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
на факультативных занятиях	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
на профориентационных занятиях	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
на занятиях с младшими школьниками	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
на внеурочных занятиях:			
игры	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
кружки по интересам и творческие проекты учащихся	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
коммерческое использование (ученический кооператив, договорные работы учащихся и др.)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
другое (что именно?)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8. Какой учебник или пособие вы используете в своей работе в качестве базового? (Укажите.)

9. Есть ли в вашем микрорайоне компьютерный клуб, центр информатики или другие возможности свободного общения с компьютером?

- да
- нет

Б. Каково ваше отношение к содержанию курса школьной информатики?

10. С какими из приводимых ниже утверждений вы согласны?

	согласен полностью	частично согласен	не согласен
Компьютерная грамотность — это прежде всего знание языков программирования	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Компьютерная грамотность предполагает прежде всего умение использовать ЭВМ для решения практических задач

Информационная культура — это прежде всего компьютерная грамотность

Изучение программирования в средней школе нецелесообразно

Программирование — вторая грамотность

Программирование целесообразно изучать лишь в профильных физ.-мат. классах и на факультативах

Информационная культура — это прежде всего культура мышления

Информатику имеет смысл изучать лишь в том случае, если имеется доступ к ЭВМ

Знание основ информатики обязательно для выпускника средней школы

Информатика должна стать со временем основополагающей, методологической дисциплиной для общеобразовательной школы

- | | | |
|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

11. Программа курса ОИВТ состоит из ряда относительно самостоятельных разделов. Какие из этих разделов вы исключили бы из обязательной программы? Какие изменили бы? Что предложили бы нового?

Исключить	Изменить	Оставить	Включить

12. Какие периодические издания вы используете при подготовке и проведении занятий?

13. Пожалуйста, назовите наиболее ценные, на ваш взгляд, научные и методические разработки (опубликованные) по преподаванию ОИВТ и использованию НИТ в образовании.

14. Каковы планируемые перспективы использования НИТ в вашей школе на ближайшие 1—2 года?

- поступление новой техники
- подготовка и переподготовка учителей информатики
- подготовка учителей-предметников к использованию ЭВТ
- компьютеризация административной или методической деятельности
- подключение к инициативному или экспериментальному проекту
- применение ЭВТ в традиционных учебных дисциплинах
- использование компьютеров в начальной и неполной средней школе
- другое (что именно?)

15. Какие затруднения испытываете вы при проведении занятий по ОИВТ или при использовании ВТ в других учебных дисциплинах? (Выберите 3—4 наиболее актуальные для вас проблемы и пронумеруйте их согласно значимости.)

- недостаточное количество ЭВТ или недостаточный доступ к ней
- ненадежность и несовершенство имеющейся техники, плохая ремонтная база
- не хватает учебных программ
- учебные программы низкого качества
- чувствую недостаточность собственной квалификации
- низкий интерес учащихся
- администрация мало уделяет внимания к проблемам преподавания информатики и использования НИТ в обучении
- не хватает методических разработок
- другое (что именно?)

Пожалуйста, сообщите еще некоторые сведения о себе:

16. Ваше базовое образование?

- педагогическое
- техническое
- инженерно-педагогическое
- другое (что именно?) _____

17. Как давно вы преподаете информатику или используете ЭВТ в учебном процессе?

	преподаёте информатику	используете ЭВТ
до 1 года	<input type="radio"/>	<input type="radio"/>
от 1 года до 3 лет	<input type="radio"/>	<input type="radio"/>
свыше трех лет	<input type="radio"/>	<input type="radio"/>

18. Вы получили специальную подготовку в области учебной информатики

- на курсах усовершенствования учителей
- в рамках базового образования
- путем самостоятельного изучения
- в хозрасчетном центре информатики, в рамках инициативного или экспериментального проекта
- другое (что именно?) _____

19. Каков ваш общий педагогический стаж?

- до 5 лет
- от 5 до 10 лет
- более 10 лет

20. Укажите ваш пол.

- муж.
- жен.

Благодарим вас за участие в опросе!

Уважаемые читатели! Мы очень надеемся, что вы примете участие в проводимом социологическом исследовании.

Ответы на анкету присылайте, пожалуйста, в редакцию по адресу: 119034 Москва, Смоленский бульвар, д. 4, издательство «Педагогика», журнал «Информатика и образование».

Экран в одном бите?

При занесении в базы данных телевизионных сюжетов (а делать это приходится все чаще) остро встает проблема размера файлов. Одна «видеосекунда» (при качественном воспроизведении цвета) занимает 20—30 Мбайт. Но поскольку большая часть этих данных дублируется (от кадра к кадру изображение меняется незначительно), можно записывать на диск и обновлять на экране только те элементы кадра, которые претерпели изменения. В комбинации с использованием известных способов сжатия неподвижного изображения это позволяет уменьшить объем файла в 100—200 раз, иначе говоря, уместить информацию, соответствующую экрану из 300 000 точек, цвет каждой из которых кодируется тремя байтами (16 миллионов оттенков!) в 5 Кбайт.



Мало того, сообщают, что в фирме Iterated Systems, основанной известным исследователем фрактальных множеств М. Барнсли, разработаны алгоритмы сжатия, позволяющие сжимать файлы с записями изображений в сотни и даже тысячи раз, причем даже те файлы, которые предварительно были сжаты «традиционными» алгоритмами, — таким образом, общий коэффициент сжатия достигает 100 000 и для записи информации об одном кадре достаточно... десяти байтов. Верить, не верить? И чего ждать дальше?

Электронный кентавр

Объединение двух разных хорошо известных предметов приводит иногда к появлению оригинальной модели со столь замечательными свойствами, что все забывают, что пользуются не новой вещью, а удачным симбиозом двух старых.

Так произошло и с ручным сканером, умеющим распознавать печатный текст, объединенным с электронным транслятором английского языка в японский.

Новое устройство, не имеющее пока короткого названия, позволяет читать английские газеты на японском языке, проводя по строчкам небольшим устройством размером со школьный пенал. Иероглифы перевода немедленно появляются на ЖК-дисплее, расположенном на верхней панели.

Очень удобно для тех, кто неважно владеет английским.

Ассоциация учителей информатики — производителям ВТ, разработчикам программных средств

Ассоциация учителей информатики — общественная организация, созданная с целью защиты прав и интересов учителя, оказания ему помощи в получении надежной ВТ и высококачественного программного обеспечения, объединяющая в своих рядах учителей, работающих более чем в 200 городах страны. Она располагает обширной информацией о техническом состоянии и программном обеспечении школьной информатики, перспективах ее развития.

Ассоциация учителей информатики приступает к формированию базы данных о производителях вычислительной техники и разработчиках программного обеспечения с целью заключения в дальнейшем договоров на приобретение их продукции.

Для включения в базу данных необходимо представить следующие сведения:

Для изготовителей ВТ

1. Наименование предприятия
2. Адрес

3. Тип производимой вычислительной техники, периферийных устройств
4. Техническая характеристика продукции
5. Дата начала выпуска
6. Примерная стоимость
7. Контактный телефон

Для разработчиков программного обеспечения

1. Автор или авторский коллектив
2. Адрес, телефон
3. Тип используемой техники (название, марка, завод-изготовитель, особенности)
4. Каталог готовых разработок с краткой аннотацией программных продуктов и их ориентировочной стоимостью
5. Перспективные разработки

Дополнительные сведения, которые вы считаете нужным сообщить.

Методические рекомендации по созданию и использованию педагогических программных средств (сборник статей)

Разработано в лаборатории средств обучения информатике, вычислительной технике и интерактивных обучающих комплексов НИИ средств обучения и учебной книги Академии педагогических наук СССР (НИИ СОиУК АПН СССР). Ответственный редактор И. В. Роберт. Ротапринт НИИ СОиУК АПН СССР. М., 1991, 106 с.

В сборнике представлены работы, в которых описываются теоретико-методологические и методические проблемы разработки и эффективного использования в учебном процессе педагогических программных средств (ППС).

Статьи посвящены вопросам педагогической целесообразности применения ППС в учебном процессе: проблемам синтеза ППС с традиционными средствами обучения; методическим рекомендациям по организации диалога пользователя с ПЭВМ; описанию

психолого-педагогических, технических требований к ППС, а также основных эргономических принципов отображения информации на экране дисплея ПЭВМ, способствующих оптимизации деятельности с ППС.

Кроме того, в сборнике содержатся материалы, обобщающие опыт создания и использования ППС в преподавании ряда общеобразовательных предметов в средней школе.

Сборник предназначен научным сотрудникам, методистам, сфере деятельности которых связана с проблемами информатизации образования, разработчикам ППС, студентам педагогических вузов; его можно рекомендовать преподавателям и учителям для использования в учебном процессе средней школы и других средних учебных заведений.

ЧТО МОЖЕТ ЭВМ

Недешево, но очень сердито

«За всё надо платить», — гласит старинная американская поговорка. В том числе, разумеется, и за удобства работы с компактными и легкими камкордерами, позволяющими так же просто, как и с помощью фотоаппарата, запечатлеть целые фрагменты жизни вашей семьи.

Но именно невесомость современных видеокамер стала причиной некачественной съемки: даже легкая дрожь руки оператора навсегда входит в историю. Конечно, можно воспользоваться штативом, но первое же землетрясение разочарует вас и в этом приборе. Есть еще один способ избавиться от сейсмической сверхчувствительности камеры — утяжелить ее, но японская фирма Panasonic выбрала другой путь.

Ее новая, по-прежнему компактная и очень легкая видеокамера NV-S1 даже в самых неуверенных руках сохраняет стабильность положения кадра и создает впечатление плавности движения рук оператора. Сгладить естественные и неестественные колебания руки позволяет цифровая обработка видеосигнала встроенным процессором.

Интересно: какие новые неудобства принесет это улучшение?

Кто кого?

Современный персональный компьютер — это на первый взгляд набор периферийных устройств. Да и стоит системная плата с процессором, памятью и микросхемами управления гораздо меньше, чем монитор, клавиатура, винчестер и принтер, взятые даже по отдельности.

И все же именно дешевизна и миниатюрность процессорных чипов привели к появлению большого числа специализированных СБИС, основное назначение которых — управление устройством, в которое они встроены. И поскольку вклад стоимости процессора в стоимость основного устройства невелик, такой периферийно-процессорный комплекс называют обычно не компьютером с переработанным внешним устройством, а, скажем, фотоаппаратом с электронным управлением.

Именно такая ситуация сложилась с плеером Walkman VM-FX90 фирмы Sony. Он имеет электронный счетчик расхода ленты, дистанционное управление, возможность воспроизведения указанного участка кассеты заданное количество раз и ЖК-дисплей для отображения всех режимов работы. Естественно, все эти удобства обеспечиваются встроенным микропроцессором, о котором в описании не идет даже речи.

А ведь еще не так давно было время, когда высококачественный магнитофон сам был одним из устройств, обеспечивающих работу центрального процессора.

Компьютерно-мозговой штурм

Для классического «мозгового штурма» необходимо несколько участников, которые «заводят» друг друга, взаимно расковывают фантазию, стимулируют генерацию идей. Так что же — один в поле не воин?! Попробуем привлечь в помощь одиночке компьютер.

Специальные программы «мозгового штурма» (их еще называют программами генерации идей) содержат наборы слов и идей, используемых для «генерации» ассоциаций; вопросники по специфическим проблемам (например, по рыночной стратегии или разработке новых изделий); модели оптимиста, пессимиста, реалиста и романтика, помогающие оценить вашу ситуацию; «морфологический ящик», перебирающий заданные ему параметры во всевозможных комбинациях (а вдруг одна из них окажется решением проблемы!), и многие другие приспособления.

Интересно, способно ли это заменить здоровый творческий коллектив...

Еще один класс ПО

Текстовый редактор, база данных, электронная таблица — что еще нужно для счастья?! Оказывается, немало: календарь, записная книжка, «напоминальник»... Конечно, дату можно узнать у операционной системы, в качестве записной книжки использовать базу данных или текстовый редактор, их же использовать для напоминания о текущих делах, но лучше бы все это интегрировать, сделать удобным для неспециализированного (уровня хозяина фирмы) пользователя, автоматизировать напоминание (а заодно составление отсортированного по времени списка дел на день) и, кстати, дать компьютеру возможность обзванивать по заранее составленному списку телефонных абонентов, передавать им информацию, запоминать в отсутствие хозяина звонивших ему...

Все это (а также множество других, мелких, но удобных компонентов) и называется персональной информационной системой. Такие пакеты все больше входят в моду на Западе.

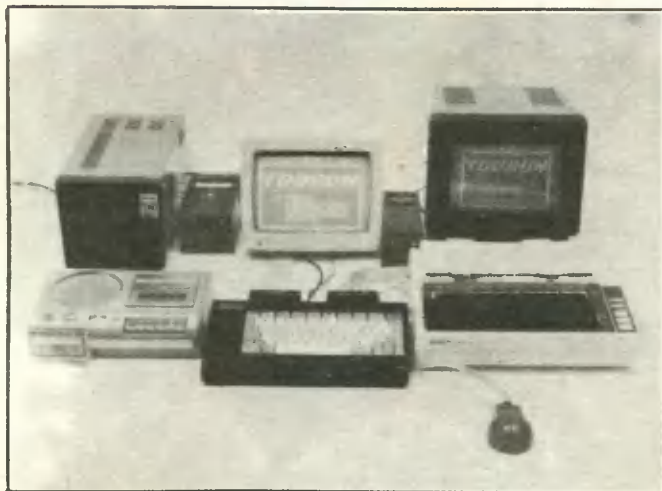
За PS/2 — PS/1

IBM тоже хочет выпускать БК — бытовые компьютеры. Однако вместо того, чтобы взять за образец БК-0010, IBM предлагает рынок вариант системы PS/2. Бытовые компьютеры PS/1 (пока четыре модели) построены на базе микропроцессора 80286, имеют ОЗУ от 512 К до 1 Мбайт, дисплей VGA, дисковод для дискет 3,5 дюйма 1,44 Мбайт, модем, могут иметь винчестер до 3 Мбайт. Компьютер продается как единый комплекс аппаратных и программных средств и услуг, избавляющий пользователей от каких-либо хлопот при эксплуатации (вот бы и наши производители так!). Главное новшество — оперативное техническое обслуживание (не более 48 ч будет требоваться для ремонта) и консультации. Цена — от 1000 до 2000 долларов; недорого, но такова уж политика IBM.

„ЭКСИТОН“

представляет семейство

„Электроника БК“



Семейство недорогих бытовых компьютеров с прекрасными возможностями «Электроника БК0010-01», «Электроника БК0011М».

Обе модели обладают полной программной совместимостью и могут объединяться в локальные сети.

Созданные на основе таких локальных сетей компьютерные классы КУВТ-86 и УКНЦ-01.01 широко применяются в школах, ПТУ, ВУЗах для изучения информатики и других предметов, а также в детских садах для обучения детей алфавиту, счету, изобразительному творчеству.

Бытовые компьютеры — хорошие помощники в работе. На Вашем рабочем столе они помогут Вам просто и удобно подготовить текстовый документ, таблицу, график, отыскать нужный материал в Вашем архиве. А дома — это прекрасное развлечение для всех членов Вашей семьи. Увлекательные компьютерные игры, самостоятельное программирование доставят Вам несомненное удовольствие.

Вашу покупку значительно удешевит то, что компьютеры «Электроника БК» могут работать не только со специализированными периферийными устройствами (видеомонитор, накопитель на гибких магнитных дисках), но и с бытовыми телевизорами и магнитофоном.

В составе программного обеспечения компьютеров — Бейсик-система, игровые программы. «Электроника БК0011М» имеет, кроме того, операционную систему с языками программирования Паскаль, Фортран, Си, средства текстообработки. Дополнительное программное обеспечение можно приобрести или заказать в ОКБ при заводе «Экситон» (т. 7-03-65 в г. Павловский Посад).

За более детальной информацией просим обращаться письменно по адресу: 142500, г. Павловский Посад, ОКБ при заводе «Экситон» или по телефонам 7-03-65, 7-03-70, 7-02-71.

Компьютеры «Электроника БК» — любопытный вариант, стоящий того, чтобы попробовать поработать с ним.

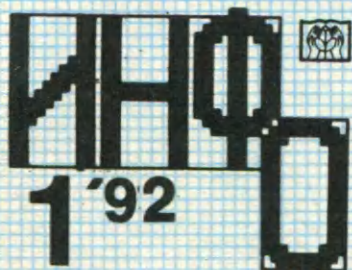


99

Цена 2 р.

Индекс 70423

OldPC.ru
7004
музей компьютеров



**ИНФОРМАТИКА
И ОБРАЗОВАНИЕ**

