

ИНФОРМАТИКА И ОБРАЗОВАНИЕ

4 1990





**НАУЧНЫЙ ЦЕНТР
ПРОГРАММНЫХ СРЕДСТВ ОБУЧЕНИЯ**

ПРЕДЛАГАЕТ

**ДЛЯ УЧЕБНЫХ КЛАССОВ УК-НЦ
САМЫЕ ПОПУЛЯРНЫЕ
СОВЕТСКИЕ И АМЕРИКАНСКИЕ
ТЕХНОЛОГИИ ОБУЧЕНИЯ ИНФОРМАТИКЕ
В ПАКЕТЕ
ПРОГРАММНЫХ СРЕДСТВ «АЛЬТЕРНАТИВА»**

- УНИВЕРСАЛЬНЫЙ СЕТЕВОЙ МОНИТОР **ВАН** – РАССЫЛКА ЛЮБЫХ ФАЙЛОВ ПО УЧЕБНОЙ ЛОКАЛЬНОЙ СЕТИ
- СИСТЕМА ПОДГОТОВКИ ТЕКСТОВ **WRITER** – РЕДАКТИРОВАНИЕ ТЕКСТОВ, БИБЛИОТЕКА ФОРМАТОВ ДОКУМЕНТОВ, ПРОЦЕССОР ПЕЧАТИ, ДРУЖЕСТВЕННЫЙ ИНТЕРФЕЙС
- УЧЕБНЫЙ ТРАНСЛЯТОР **MIT-LOGO** – ГРАФИКА, ВЫЧИСЛЕНИЯ, СЛОВА И СПИСКИ, МУЗЫКА, ПЕЧАТЬ. В ВАШИХ РУКАХ ВСЕ ВОЗМОЖНОСТИ **СЕЙМУРА ПЕЙПЕРТА!**
- ИНТЕГРИРОВАННАЯ СИСТЕМА ТЕКСТООБРАБОТКИ **МИКРОМИР**
- УЧЕБНАЯ СРЕДА ПРОГРАММИРОВАНИЯ В КОДАХ **ДЕМИР**
- ИГРОВОЙ КЛАВИАТУРНЫЙ ТРЕНАЖЕР **КЛАВИР**
- ПРОСТЕЙШАЯ УЧЕБНАЯ СРЕДА **АЛЬФАМИР**
- СИСТЕМА **КУМИР** ДЛЯ ШКОЛЬНОГО АЛГОРИТМИЧЕСКОГО ЯЗЫКА
- ИНТЕРФЕЙС **ИНФОМИР**

**КАЖДАЯ ПРОГРАММА ПОСТАВЛЯЕТСЯ В КОМПЛЕКТЕ
С МЕТОДИЧЕСКОЙ ЛИТЕРАТУРОЙ**

**ПРИНИМАЮТСЯ ЗАКАЗЫ
НЕ МЕНЕЕ, ЧЕМ НА 10 ПАКЕТОВ**

**ЗАКАЗЫ
НА ОТДЕЛЬНЫЕ ПРОГРАММЫ! ПАКЕТА
НЕ ПРИНИМАЮТСЯ**

**ЦЕНА ПАКЕТА: В ЗАКАЗЕ ОТ 10 ДО 100 – 2500 Р
ОТ 100 ДО 1000 – 1500 Р
СВЫШЕ 1000 – 1000 Р**

ЗАКАЗЫ НАПРАВЛЯТЬ ПО АДРЕСУ:

**121002, МОСКВА, СЕРЕБРЯНЫЙ ПЕР., Д.2
ТЕЛЕФОН: 290·10·49, 290·10·03**





ИНФОРМАТИКА И ОБРАЗОВАНИЕ

OldPC.su

7 0 0 3

музей компьютеров

Содержание

Общие вопросы

| | |
|--|----|
| Уваров А. Информатика в школе: вчера, сегодня, завтра | 3 |
| Когдов Н., Цевенков Ю. Центры новых информационных технологий: цели и задачи | 11 |
| Садовникова Н., Ратинский М. «Информатика» — базовый предмет в школах РСФСР | 15 |

Методика обучения

| | |
|--|----|
| Гольцман М., Дуванов А., Зайдельман Я., Первин Ю. Исполнители | 17 |
| Каптелинин В., Дроздова Т. Опыт обучения работе с текстовым редактором | 26 |
| Акимова Г., Деца В. Демонстраторы учат программированию | 32 |
| Урнов В., Зубченко А. Переходные среды в предмете «Информатика» | 36 |

Кабинет ВТ

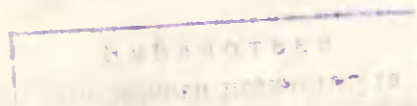
| | |
|---|----|
| Григорьев С. Работа системы Пролог-Д | 41 |
| Горбунов В. Сравнение текстов по моделям позиций слов | 45 |
| Караваев А. Цветной демонстрационный монитор | 49 |
| Ходаков П. Спасение файлов на «Агате» | 50 |
| Сабиров Р. Простой клавиатурный тренажер | 53 |
| Шкребец Ю. Неожиданное применение | 54 |
| Программа печатает сама себя (компьютерный аукцион) | 54 |
| Гельтищева Е., Селехова Г. Как предупредить переутомление при работе с видеотерминалом? | 55 |
| О доплате за заведование кабинетом вычислительной техники и за работающие компьютеры | 57 |
| Возвращаясь к теме... | 58 |

Клуб БК

| | |
|--|----|
| Зальцман Ю. Архитектура и программирование на языке ассемблера БК-0010 | 61 |
| Панченков И. Приручение БК | 68 |

Педагогический опыт

| | |
|--|----|
| Караваева Э. Нетрадиционные формы обучения на уроках информатики | 81 |
| Санжаров Л., Финьков А. Компьютер, цвет, язык | 83 |
| Караев Ж., Рах С. Использование компьютеров в казахской школе | 86 |



Внеклассная работа

«Уральские» задачи

88

Педагогические кадры

Проблемы информатизации: взгляд из педвуза

95

НИТ в дошкольном образовании

Горвиц Ю. Развивающие игровые программы для дошкольников

100

Молодежная инициатива

107

Точка зрения

Лукьянов А., Салимжанов Р., Ситников В. О модели выпускника средней школы по курсу ОИВТ

113

Нам пишут

Уважаемые коллеги!

117

Предлагаем всем

119

Как быть?

119

Настоятельно просим

120

Информация

Ассоциация создана

123

Дубна-89

126

Семинар на ВДНХ

127

Главный редактор
академик

В. А. МЕЛЬНИКОВ

Редакционная
коллегия

И. Н. АНТИПОВ

В. Н. АФАНАСЬЕВ

И. М. БОБКО

Г. В. ГОДЖЕЛЛО

С. А. ЖДАНОВ

Б. В. ЛОМОВ

Ю. В. ЛУИЗО

(зам. главного
редактора)

Н. Г. МЕЛЬДИАНОВ

И. С. ОРЕШКОВ

О. К. ПАВЛОВА

А. Ю. УВАРОВ

А. И. ФУРСЕНКО

В. О. ХОРОШИЛОВ

К. В. ШЕХОВЦЕВ

(редактор отдела)

Обложка Э. Бажилина, В. Соломонова

Редактор отдела А. Кравцова

Научный редактор Т. Драгныш

Зав. редакцией Н. Игнатова

Художественный редактор Л. Розанова

Корректор О. Пурлова

Сдано в набор 25.05.90. Подписано в печать 02.07.90. А 06 393.
Формат 70×100 1/16. Бумага офсетная № 1. Печать офсетная.
Усл. печ. л. 10,40. Усл. кр.-отт. 42,88. Уч.-изд. л. 12,95.
Тираж 72 210 экз. Заказ 1032. Цена 60 коп.

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

Адрес для переписки: 107005, Москва, Лефортовский пер., 8.

Адрес редакции: Студенческая ул., 19, корп. 1, кв. 17.

Телефон редакции: 249-97-77.

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

© «Педагогика», «Информатика и образование», 1990.

А. УВАРОВ

Информатика в школе: вчера, сегодня, завтра

Проблема обучения информатике в общеобразовательной школе продолжает вызывать многочисленные споры, особенно острые сегодня, когда разворачивается процесс глубокой перестройки всей образовательной системы.

Цель статьи — дать по возможности целостный взгляд на проблему, отражающий динамику неизбежного в предстоящие десятилетия процесса коренной перестройки содержания, методов и организационных форм обучения в школе, а также связанного с этим изменения роли и места изучения информатики.

Автор надеется, что изложенный подход может стать отправной точкой для всестороннего обсуждения путей развития содержания образования, для совместной работы всех, кто способствует развитию процесса информатизации в нашей школе.

Информатизация общества и место информатики в системе общего образования

Информатизация общества — результат проникновения в повседневную жизнь достижений научно-технического прогресса. Она стимулирует изменение содержания, методов и организационных форм обучения. Ведущим в этом процессе является изменение целей и содержания образования, технологическое же переоснащение является производным.

На этапе индустриализации общества

система образования была ориентирована в первую очередь на подготовку дисциплинированных грамотных исполнителей. В условиях информатизации «подготовка рабочей силы» в старом ее понимании перестает выступать в качестве доминирующей задачи образования. Основной упор переносится на общую гуманитарную подготовку и формирование у учащихся последовательного естественнонаучного представления об окружающем мире. 3

Информатизация образования — длительный и противоречивый процесс, который будет разворачиваться на протяжении десятилетий. В его развитии можно выделить ряд последовательных этапов. На первом происходит осознание обществом объективной необходимости и сути процессов информатизации, начинается массовое освоение технических средств НИТ (прежде всего ЭВМ), разворачивается исследовательская работа по их педагогическому освоению. Ядром этого этапа в школе является курс информатики, служащий полигоном для практического освоения школой динамично изменяющегося содержания образования, нетрадиционных методов и организационных форм учебной работы, технических средств НИТ.

По мере развития технической базы инфосферы начинается второй этап. Его основное содержание: продолжение процессов первого этапа для вновь появляющихся средств НИТ; внедрение ВТ в

традиционные учебные дисциплины и на этой основе массовое освоение педагогами новых методов и организационных форм учебной работы; практическая постановка вопроса о радикальном пересмотре содержания образования, традиционных форм и методов учебно-воспитательной работы; разработка и начало освоения «компьютерных курсов», представляющих собой законченные системы учебно-методического обеспечения.

4 Характерной особенностью второго этапа становится развертывание внутришкольных автоматизированных информационно-управляющих систем, поддерживающих все многообразие возможных форм организации учебно-воспитательного процесса. Системы образования некоторых развитых стран после десятилетних усилий уже приближаются ко второму этапу.

На третьем этапе, вероятно, произойдет радикальная перестройка содержания образования в массовой школе, смена методической основы и пересмотр организационных форм обучения.

Очевидно, что роль и место изучения информатики в школе на каждом этапе будут существенно меняться. Представляется оправданным рассматривать процесс информатизации образования лишь для первого и начала второго этапа.

Пройденный путь

Предысторию изучения информатики в школе отсчитывают с 1959 г., когда молодой учитель (ныне член-корреспондент АПН СССР) С. И. Шварцбург начал в 444-й школе Москвы успешный эксперимент по обучению старшеклассников работе на ЭВМ.

Качественно новый характер изучение информатики приобрело после принятия в 1985 г. решения об обеспечении компьютерной грамотности учащихся в широком использовании ЭВМ в учебном процессе. Основными задачами тогда были: обеспечение общеобразовательной подготовки всех учащихся средних учебных заведений в области информатики, предпрофессиональная подготовка определенной части молодежи в области ВТ, ознакомление работников школы с возможностями, предоставляемыми НИТ.

Массовому появлению в учебных заведениях современных ЭВМ было предпослано введение курса «Основы информатики и вычислительной техники», что помимо решения указанных выше задач должно было подготовить школу к быстрому освоению и эффективному использованию ВТ, массовое производство которой подготавливалось в это время, и создать условия для развертывания следующего этапа (обновление методов освоения учащимися традиционного содержания образования), а в последующем — для кардинального пересмотра содержания и перестройки всей организации учебно-воспитательного процесса.

Содержание курса ОИВТ. Источниками содержания курса ОИВТ явились практика обучения школьников основам программирования и представление о компьютерной грамотности как составной части общего образования. На этапе введения курса средства ВТ в школах практически отсутствовали, поэтому учебно-методические материалы предусматривали в качестве временной меры возможность изучения его без ЭВМ. В этом контексте основной упор делался на алгоритмизацию как средство развития специфических способностей, дополняющих традиционно формируемые в рамках курса математики.

В практике работы школы реальное содержание курса определялось наличием опытом педагогов, уровнем и содержанием их дополнительной подготовки; набором фактически доступных технических и программных средств; характером доступной учителям информации о мировой практике использования ЭВМ.

В силу методической традиции новый курс фактически оказался внутренне связан с курсами по программированию, а призыв к овладению компьютерной грамотностью часто трактовался как призыв к овладению приемами вычислений на ЭВМ. В ходе становления содержания курса ОИВТ реализовывались различные, часто противоположные точки зрения: его ориентировали на использование микрокалькуляторов; делали акцент на организацию решения задач с использованием ЭВМ, на развитие логического мышления и формирование у

школьников приемов доказательства правильности программ; сливали курс с системой предпрофессиональной подготовки в области программирования; пытались ориентировать его на освоение учащимися программных средств массового применения; предлагали превратить в общеметодологическую дисциплину и т. п. Однако для многих учащихся изучение информатики даже и сегодня сводится к знакомству с конструкциями Фокала или Бейсика, реализацию с их помощью простейших алгоритмов.

В условиях, когда содержание курса во многом остается подвижным, разработчики учебно-методических материалов и учителя по-прежнему делают основной упор на развитие тех его составляющих, которые им лучше знакомы и для которых, по их мнению, проще организовать обучение, используя имеющиеся возможности. Вместе с тем интенсивное развитие методических работ, подготовка современного учебника сформировали новую устойчивую тенденцию, о которой пойдет речь ниже.

Технические средства. В 1985 г. было решено начать массовый выпуск персональных ЭВМ для оснащения учебных заведений. Однако это решение практически не выполняется. Имеющаяся техника крайне ненадежна. Значительная часть установленных в школах машин по своим техническим и физиолого-гигиеническим характеристикам не удовлетворяет требованиям учебного процесса. Школы не обеспечены машинными носителями информации, плоха организация обслуживания и ремонта учебной ВТ.

Опыт использования в ряде учебных заведений доброкачественной импортной ВТ показывает, что отсутствие технических проблем и сравнительное обилие программного обеспечения необходимы учителям, чтобы вплотную приблизиться к использованию огромного методического потенциала ПЭВМ.

Отсутствие удовлетворительных средств ВТ — главная причина, тормозящая развертывание первого этапа информатизации образования.

Учебно-методическая литература и программное обеспечение. Основу изучения курса информатики заложили раз-

работанное в сжатые сроки пробное учебное пособие и сопровождавшие его методические материалы. Вслед за ними появилось более сотни дополнительных учебных и методических пособий, посвященных обучению работе с микрокалькуляторами, программированию на языках Бейсик, Фокал и т. п. Пособия, как правило, обобщали опыт обучения программированию применительно к тем или иным конкретным условиям. Впоследствии стали издаваться лучшие из современных учебных и учебно-методических пособий, подготовленные на достаточно высоком уровне.

К сожалению, не разрешен пока один из главных вопросов — создание индустрии педагогических программных средств. Обучение информатике является первой предметной областью, для которой уже сегодня есть реальная возможность создания законченного компьютерного курса, однако необходимая централизованная поддержка разработчиков отсутствует.

Педагогические кадры. Подготовка учителей — определяющий фактор успешного становления обучения информатике в школе. Сейчас основную массу преподавателей информатики составляют учителя математики и физики, прошедшие краткосрочную подготовку. Иногда информатику ведут специалисты по ВТ и программированию, не имеющие педагогического образования.

Краткосрочные курсы не дали учителям необходимых знаний и навыков работы с ЭВМ. В результате многие ограничиваются формальным пересказом содержания учебника, а практические занятия на ЭВМ сводят к составлению простейших программ.

Не закончен процесс становления курсов по подготовке учителя информатики и в педагогических вузах. Ощущается нехватка педагогов высшей квалификации для развертывания этой работы. Остро недостает учебных ЭВМ, необходимых педагогам в первую очередь.

Организационные методы и формы обучения. Курс информатики дает уникальные возможности по разнообразию форм и методов работы с учащимися. ЭВМ — замечательный инструмент для индивидуализации обучения. Однако на

практике этот потенциал редко используется полностью. Соответствующие методические разработки практически отсутствуют. Слабо обобщен опыт учителей, организующих обучение по методу проектов, позволяющему гибко сочетать работу школьников на уроке с самостоятельной внеклассной работой. Практически не изучены возможности самостоятельной работы школьников с обучающими программами, тренажерами.

Значительный объем лабораторных занятий обеспечивает в рамках курса благоприятные предпосылки для формирования атмосферы сотрудничества, сокращения дистанции, существующей между педагогом и учащимися в традиционных учебных ситуациях. Однако нужно многое еще сделать, чтобы открывающиеся здесь возможности стали достоянием каждого педагога.

Вариативность обучения. Условия, в которых ведется обучение информатике, зачастую существенно различны не только в разных регионах страны, но и в рядом расположенных школах. В результате выбор технических средств, программного и учебно-методического обеспечения, способов организации занятий определяется возможностями, имеющимися на местах, а ход и результаты обучения варьируют в самых широких пределах. Недостает единых требований, позволяющих сопоставить эффективность работы различных школ.

К настоящему времени накоплен значительный опыт внеклассной работы школьников по информатике, формируются факультативные курсы. Складывается сеть региональных школ юных программистов, организуются летние лагеря и конференции школьников по информатике. Однако развитие углубленной и предпрофессиональной подготовки школьников сдерживается нехваткой высококвалифицированных педагогов, ВТ и расходных материалов, недостаточной разработанностью соответствующих курсов. Становление системы предпрофессиональной подготовки в области информатики продолжается.

Отношение к курсу ОИВТ. Его введение было полной неожиданностью для большинства учащихся и педагогов. В результате предпринятых усилий и разъ-

яснительной работы большинство педагогов приняли необходимость нового курса и начали осваивать его содержание. Общественность в основном поддержала новый курс, который рассматривался как символ возможных глубоких изменений в условиях начинавшейся реформы школы. Учащиеся с энтузиазмом встретили возможность познакомиться с практически недоступной для большинства из них ВТ в школе, но затем их отношение к курсу стало меняться.

Школьники встретились на уроках информатики с чем-то внешне очень близким к традиционным математическим дисциплинам. Практическое знакомство со средствами НИТ для большинства из них заменялось в лучшем случае передачами учебного телевидения. Падение интереса вместе с другими факторами отразилось на результативности обучения. По данным НИИ СиМО АПН СССР, в 1988 г. лишь один из пяти учащихся полностью освоил курс ОИВТ.

В школах, где есть квалифицированные педагоги и где учащиеся имеют достаточный доступ к ВТ, их исходное положительное отношение к предмету получило развитие. Здесь наблюдается повышенный интерес к профессиям, связанным с использованием ЭВМ, высок общий уровень освоения содержания курса. Умение решать задачи с помощью компьютера столь же престижно для школьников, как победы в компьютерных играх. Выражения из области информатики становятся заметной частью школьного фольклора, а компьютеры — темой для общения учащихся. Появление выпускников средней школы, имеющих необходимую подготовку в области информатики, начали ощущать преподаватели высшей школы.

Объем курса информатики невелик, но требует от педагогов большой работы по его начальному освоению. Много трудностей вызывают получение и установка компьютеров, поддержание их в работоспособном состоянии. Еще больших усилий требуют поиск и освоение педагогических программных средств. Высокая трудоемкость курса, не вполне определенные перспективы его развития влияют на отношение учителей к этому предмету. Вместе с тем престиж квалифици-

рованных преподавателей информатики достаточно высок и их нехватка ощущается повсеместно.

Назревшие задачи

Обучение информатике в общеобразовательной школе переживает кризис. Безмашинный вариант себя изживает. Новый компьютерный курс информатики, формирующий представление о становлении информационного общества и его ценностях, ориентированный на обучение учащихся основам алгоритмизации, общим методам решения задач, еще только получает права гражданства. Параллельно выявляются глубинные связи обучения информатике с другими учебными предметами, с гуманитаризацией образования. Осознается необходимость переноса курса в неполную среднюю школу. Все это происходит в условиях начавшейся радикальной перестройки системы общего образования.

Сегодня вопрос о роли и месте обучения информатике определяется ожидаемой динамикой процессов информатизации нашего общества, перестройкой содержания и методов учебно-воспитательной работы школы, накопленным опытом преподавания курса.

Главными особенностями современного этапа являются:

усиление общеобразовательных аспектов содержания обучения информатике; переход к интенсивному использованию ЭВМ в курсе ОИВТ и применению их при изучении других общеобразовательных дисциплин;

понижение возрастной границы начала обучения;

варьирование содержания обучения, видов и способов работы с ЭВМ для учащихся старших классов.

Обучение информатике (как и другим предметам) решает две группы задач. Одна из них связана с формированием у школьников общей научной картины мира, другая — с формированием специфического набора конкретных знаний, умений и навыков.

Решение первой группы задач предполагает, что школьникам надо дать представление об информационных процессах в природе и обществе, интегративном характере информатики и соот-

ношении ее с другими дисциплинами, информатизации общества как закономерной стадии его развития, о соответствующих этой стадии этических нормах.

Мировоззренческая составляющая содержания обучения информатике еще только формируется в общественном сознании и занимает пока незначительное место в учебных программах. Фундаментальный характер информатики, ее влияние на современное научное мировоззрение раскрываются недостаточно. Это положение будет меняться вместе с соответствующими изменениями в окружающей нас инфосфере (развитие глобальных систем телекоммуникации, массовое распространение профессиональных ЭВМ, построение систем, затрагивающих повседневную жизнь существенной части граждан нашей страны: «электронные деньги», глобальные и локальные базы данных и т. п.), ростом числа и престижности профессий, требующих высшей квалификации, и нарастанием соответствующих изменений в общественном сознании.

Другая составляющая обычно описывалась метафорой «компьютерная грамотность». Компьютерную грамотность в узком смысле по аналогии с обычной грамотностью часто трактуют как умение «читать» (знакомиться с текстами, хранящимися на машинных носителях информации, в базах данных и т. п.), «писать» (готовить и распространять тексты с помощью ЭВМ), «вычислять» (пользоваться электронными таблицами и т. п.). В более широком смысле она включает умение составлять алгоритмы, организовывать решение возникающих задач (планирование действий, их расчленение на составляющие и выстраивание в последовательность для достижения желаемого результата, формирование навыков систематически разбивать задачи на подзадачи, организовывать совместную работу исполнителей); умение самостоятельно осваивать готовые программные средства ЭВМ.

Задача обеспечения компьютерной грамотности в узком смысле слова возникает на начальном этапе информатизации общества. В дальнейшем все большее значение начнет приобретать задача обеспечения компьютерной грамотности

в широком смысле слова. Ее решение является важным элементом формирования в обществе информационной культуры как составной части культуры, необходимой для полноценной жизни в технологизированном, быстро меняющемся мире.

Возрастное распределение. Современный подход, при котором изучение курса ОИВТ ведется в старших классах параллельно с углубленной подготовкой школьников и курсами профессиональной подготовки в области ВТ, себя исчерпал. Предстоит разделить по времени первичное ознакомление с ЭВМ, изучение общеобразовательного материала и профессиональную (предпрофессиональную) подготовку.

8 Опыт показывает, что первоначальное знакомство с ЭВМ может с успехом осуществляться на этапе начального обучения. Одним из наиболее продвинутых экспериментов в этой области сегодня является обучение младших школьников на базе системы «Роботландия».

Год назад общедоступной стала русскоязычная версия Лого. Она может стать хорошей основой для разработки системы развивающего обучения младших школьников с использованием ЭВМ.

Базовую общеобразовательную подготовку по информатике естественно проводить в неполной средней школе, углубленную и предпрофессиональную подготовку — в старших классах.

На первом и в начале второго этапа информатизации образования информатика может выступать в виде отдельного учебного предмета. По мере интеграции учебных дисциплин ее место в системе учебных предметов должно уточняться.

Содержание общеобразовательной подготовки вычлняется из поля профессиональной деятельности в области информатики по мере того, как ее составляющие начинают приобретать общезначимый характер. Ознакомление учащихся с работой на ЭВМ имеет переходящее значение. Быстрое совершенствование ВТ и программных средств, «дружественный интерфейс» и т. п. позволяют даже неподготовленному пользователю быстро осваиваться с новыми техническими и программными средствами. Отпадает необходимость в специальной подготов-

ке к использованию ЭВМ и программных средств общего назначения в школе. Основные усилия предстоит сосредоточить на решении мировоззренческих задач, на формировании компьютерной грамотности в широком ее понимании (впрочем, не снимается и задача выработки навыков работы с клавиатурой).

Ядро базовой подготовки по информатике может составить освоение учащимися новой для них знакомой системы, представленной алгоритмическим языком. Она отличается от знаковых систем традиционной школьной математики своей конструктивностью, возможностью ее «материализации» с помощью ЭВМ.

По-видимому, наиболее проработанным и учитывающим современное состояние методических разработок в этой области является сегодня вариант содержания общеобразовательного курса «Информатика», предложенный А. П. Ершовым, А. Г. Кушниренко, Г. В. Лебедевым, А. Л. Семеновым, А. Х. Шенем. Этот вариант ориентирует учащихся в потоке событий, ведущих к становлению информационного общества, закладывает основу «алгоритмического мышления», эффективного использования быстро развивающихся информационных технологий.

Содержание обучения информатике в старших классах может (в зависимости от имеющихся возможностей, специализации школы и пожеланий учащихся) включать углубленное изучение отдельных вопросов базового курса и других пользующихся популярностью тем (например, основы теории алгоритмов, машинное моделирование, вычислительная математика, информационные системы, методы искусственного интеллекта) по выбору учащихся, предпрофессиональную и профессиональную подготовку школьников (автоматизированное делопроизводство, программирование и т. п.). Ведущая роль здесь, по-видимому, будет принадлежать межшкольным кабинетам, центрам информатики и т. п.

Появление в школах ЭВМ позволит развернуть факультативные занятия по предметам естественнонаучного и гуманитарного профиля, предусматривающие интенсивное использование ЭВМ, овла-

дение специальными программными средствами (автоматизированный эксперимент, построение и/или изучение моделей процессов и явлений, машинная графика, музыкальная композиция и т. п.).

Качество и ассортимент технического и программного обеспечения — критические на сегодня факторы развития обучения информатике. Они определяют методы и организационные формы, а зачастую и содержание обучения.

Как показывает опыт, дешевые бытовые компьютеры пригодны только для первоначального знакомства школьников с ЭВМ. Техническую основу обучения информатике должны составлять типовые ЭВМ с достаточно высокими характеристиками (за рубежом все шире используются профессиональные ПЭВМ) и расширенным ассортиментом периферийного оборудования (механические исполнители, датчики, лабораторное оборудование). Особое внимание стоит уделить средствам телекоммуникации, которым предстоит сыграть важную роль на втором этапе информатизации образования. Безвредность для здоровья и надежность остаются главными условиями, которым должны удовлетворять школьные компьютеры.

В наступающем десятилетии будет быстро возрастать роль видеотехники в учебном процессе. В перспективе компьютеры и видеосредства будут интегрированы в единое обучающее окружение. Первые образцы такого интегрированного окружения уже эксплуатируются за рубежом. Нам предстоит оценить и освоить их дидактический потенциал. Для нормального изучения информатики необходим специально разработанный комплект программного обеспечения, позволяющий поддерживать каждую рассматриваемую тему и включающий сетевую операционную систему, инструментальные программные средства (интерактивный экранный ассемблер-деассемблер, редакторы-компиляторы с языков Си, Паскаль, Бейсик), программные средства общего назначения (редакторы текстов, графический и музыкальный, база данных, электронные таблицы, текстовый процессор с анализом грамматической правильности

текстов, редакционно-издательская система), программные средства, поддерживающие изучение информатики в начальных, средних и старших классах школы. Создание и распространение этих средств — самая актуальная сегодня задача. Все они должны отвечать требованиям, предъявляемым к программным средствам массового применения, обеспечивать достаточную производительность, иметь единый пользовательский интерфейс.

Подготовка педагогических кадров — ключевое звено успешного развития обучения информатике. Сегодня подготовка большинства учителей недостаточна для полноценного преподавания ОИВТ. При подготовке к переводу курса в неполную среднюю школу каждый учитель информатики должен получить возможность пройти занятия по программе, по которой будут работать его учащиеся. Переподготовка должна проводиться в образцово оборудованном кабинете информатики с полным использованием дидактического потенциала учебной вычислительной техники.

Предстоит перестроить подготовку будущих учителей информатики в педагогических вузах. Предлагается пересмотреть состав дисциплин цикла «Информатика», включив в их состав следующие:

введение в информатику и практическое использование ЭВМ. Этот курс носит пропедевтический характер. По мере повышения уровня обучения информатике необходимость в нем отпадет;

теоретическая информатика. Сюда входят два обычно не связываемых между собой раздела: элементы математической логики, теория алгоритмов и вопросы теоретического программирования; информационные системы и их место в обществе, документалистика;

прикладное программирование: введение в операционные системы ЭВМ; знакомство с производственными языками и современной технологией программирования; основы вычислительной математики и численные методы решения задач; введение в машинную графику;

вычислительная практика: выполнение самостоятельного программистского

проекта, освоение разработки программных средств размером в тысячу строк; информационная технология в учебном процессе: методика проектирования, разработки, оценки и использования программных средств в учебном процессе;

методика преподавания информатики.

Управление развитием обучения информатике. Опыт становления обучения информатике показывает, что выделяемые на это средства и затрачиваемые усилия часто не дают желаемых результатов. Научно-методическое обеспечение проводимых работ совершенно недостаточно. Отсутствует эффективный мониторинг развивающейся практики внедрения ЭВМ в образование.

10 Для коренного улучшения дел необходима перестройка практики финансирования и координации разработок, подготовки и переподготовки педагогических кадров, оснащения учебных заведений средствами ВТ. Необходимо конкретные программы развития — как глобальные, финансируемые из централизованных источников, так и региональные, базирующиеся на местных ресурсах.

Поддержка коллективов, добившихся опережающего внедрения информатики в образование, развитие исследовательских проектов, создание прообразов учебных заведений завтрашнего дня — главная задача централизованных программ. Основная цель местных программ — обеспечить повышение уровня работы конкретных учебных заведений, используя НИТ там, где это приносит максимальный педагогический эффект.

Решение об обеспечении компьютерной грамотности учащихся и широком использовании ЭВМ в учебном процессе вывело общеобразовательную школу в лидеры процесса информатизации нашего общества. Вместе с тем ясно, что ресурсов для обеспечения фронтального решения этой задачи сегодня недостаточно. Школе, как и обществу в целом, не

хватает технических средств НИТ; нет индустрии программного обеспечения; отсутствует необходимый научно-методический задел, не хватает специалистов для подготовки и проведения опережающих исследований; недостаточно подготовка учителей.

В этих условиях целесообразно отказаться от изучения информатики в тех школах, где отсутствуют необходимые материальные условия и квалифицированные кадры; изучать курс факультативно там, где у квалифицированных педагогов нет возможности организовать практическую работу учащихся на ЭВМ; разрешить квалифицированным педагогам, ведущим занятия в кабинетах ВТ, широко экспериментировать с учебной программой, учебниками, организационными формами и методами проведения учебных занятий.

Необходимо интенсифицировать экспериментальные исследования, подготовку программного и методического обеспечения для перенесения курса ОИВТ в средние классы уже в ближайшие годы.

Технические средства НИТ — основа информатизации образования. Необходимо отказаться от валовых показателей оснащения школ учебной ВТ, выпускать ее в объемах, позволяющих промышленности поддерживать высокий уровень качества продукции. Надо признать, что разрыв в уровнях учебной работы между различными школами будет возрастать, и использовать передовые учебные заведения для отработки качественно новых форм и методов учебной работы.

Компьютеры сами по себе не несут педагогических идей, но дают возможность реализовать наработанные в педагогике концепции, которые не были осуществлены из-за отсутствия необходимых средств. ЭВМ являются средством перестройки образования, а сама перестройка — необходимое условие их эффективного использования.

Н. КОГДОВ,

зав. отделом Общесоюзного центра НИТ

Ю. ЦЕВЕНКОВ,

зав. сектором Общесоюзного центра НИТ

Центры новых информационных технологий: цели и задачи

Информатизация образования — ключевое условие информатизации общества — достаточно сложный и длительный процесс, связанный с развитием научных исследований, техническим оснащением учебных заведений, подготовкой и переподготовкой педагогических кадров и проведением целого ряда организационных мероприятий, направленных на создание инфосферы информатизации.

Создание и утверждение «Концепции информатизации образования» позволило определить основные направления информатизации народного образования. К числу таких направлений в первую очередь следует отнести вопросы создания и развития инфраструктуры информатизации.

Основу такой инфраструктуры должны составлять созданные на базе ведущих научных организаций и вузов Общесоюзный центр НИТ, центры НИТ по уровням образования, республиканские и региональные центры НИТ. Важным ее элементом должны стать создаваемые ассоциации вузов, молодежные студенческие объединения и центры НТТМ, научно-исследовательские объединения совместных предприятий и кооперативов, научно-методические объединения, ИПК, ФПК, отраслевые и региональные фонды алгоритмов и программ.

При функционировании такой структуры наряду с госбюджетными ассигнованиями в необходимых случаях должны использоваться принципы хозрасчета и самоокупаемости. Для обеспечения ее работоспособности необходимо создание и использование системы распределенных баз данных и знаний, электронной почты, средств организации телеконференций и т. д.

При этом должны быть разработаны организационно-экономический механизм информатизации образования, методы управления этим процессом, си-

стемы правовых и организационно-методических документов, обеспечивающих интенсификацию процессов информатизации, их развитие на базе экономических методов управления.

В системе центров НИТ необходимо предусмотреть создание экспертных комиссий по диагностике и тестированию разработанных программных средств с предоставлением этим комиссиям права присвоения аттестованному средству статуса учебника, учебного пособия, учебной программы.

Развитие и управление такой системой осуществляется Общесоюзным и уровневыми центрами НИТ под руководством Гособразования СССР.

В настоящее время создан и начал практическую деятельность Общесоюзный центр новых информационных технологий на базе НИИ проблем высшей школы.

Основными задачами Общесоюзного центра являются:

проведение единой научно-технической и учебно-методической политики в области информатизации образования в системах непрерывного образования СССР и в рамках программ международного сотрудничества в странах СЭВ;

участие в разработке прогнозов развития программного и технического обеспечения учебного процесса в системе образования; определение наиболее перспективных программных средств учебного назначения и методов использования вычислительной техники в соответствующих направлениях работ;

разработка перспективных планов и программ научно-исследовательских работ в области создания методов и средств новых информационных технологий обучения;

исследование эффективности использования инструментальных программ учебного назначения;

координация научных исследований в области новых информационных технологий, внедрения вычислительной техники в учебный процесс; создание, тиражирование и распространение программных средств учебного назначения;

координация разработки и выпуска средств учебно-методического и программного обеспечения новых информационных технологий;

организация освоения и эксплуатации программных и технических средств, поступающих в систему образования;

накопление и систематизация общего и прикладного программного обеспечения для работы на вычислительной технике (в том числе и зарубежной);

создание Общесоюзного информационного фонда по учебно-методическому и программному обеспечению НИТ. Обеспечение разработчиков и потребителей информацией о разработках программного обеспечения нового учебного назначения;

организация и осуществление разработки отраслевых нормативов и стандартов в области новых информационных технологий обучения;

организация конкурсов на разработку программно-методического обеспечения учебного процесса;

экспертиза разрабатываемых программных средств учебного назначения; обобщение и анализ информации о разрабатываемых в стране и за рубежом методах и средствах новых информационных технологий обучения;

анализ оснащенности учебных заведений вычислительной техникой и разработка предложений по ее распределению;

анализ и оценка эффективности применения программно-методического обеспечения новых информационных технологий в учебном процессе;

разработка предложений по распределению финансовых и материальных средств на проведение научно-исследовательских и опытно-конструкторских работ по созданию программного и методического обеспечения учебного процесса для всех уровней образования и представление их в Госкомитет СССР по народному образованию;

повышение квалификации руководителей системы народного образования и

специалистов в области новых информационных технологий обучения.

ОЦ НИТ должен работать в тесном контакте с региональными центрами новых информационных технологий, оказывать содействие в освоении новых информационных технологий, активно пропагандировать и создавать условия для освоения и использования средств вычислительной техники.

Сформированы и начали свою практическую деятельность 12 региональных центров новых информационных технологий на базе вузов различных регионов страны. Около 20 вузов подготовили предложения о создании на их базе центров новых информационных технологий, которые находятся в стадии согласования с головной организацией.

Создание организационно-экономического механизма информатизации и соответствующей структуры даст возможность организовать проведение целевых комплексных научных исследований по следующим направлениям:

теоретико-методологические проблемы информатизации в сфере образования и педагогической науки, формирование концептуальных подходов и методологий информатизации системы народного образования;

содержание, организация и методы изучения информатики и других дисциплин в области информатизации;

проблемы разработки и применения новых информационных технологий в сфере образования;

создание учебно-методического и программного обеспечения учебного процесса, изменение содержания подготовки на основе информационных технологий; психолого-педагогические проблемы информатизации образования;

анализ и прогнозирование социальных аспектов информатизации сферы образования, социально-экономический анализ компьютеризации школы;

научно-технические, эргономические и социально-экономические основы создания технических средств информационных технологий и новых поколений вычислительных систем для сферы образования;

физиолого-гигиенические проблемы применения ЭВМ в условиях информа-

тизации общества;

психолого-педагогические и методические основы применения ЭВМ в дошкольных учреждениях;

проблемы создания и применения ЭВМ и других средств информатики для людей с физическими недостатками;

создание структуры и состава технического обеспечения процесса обучения (классы, лаборатории, стенды и т. д.);

разработка дидактически обоснованных моделей и методов автоматизации труда учащихся в системе среднего и профессионального образования;

исследования по составу и разработка базовых, инструментальных и прикладных программных средств для учебных ПЭВМ;

разработка системы тренажеров, имитаторов и технических комплексов на основе НИТ для подготовки квалифицированных рабочих сложных профессий;

создание аппаратно-программной поддержки курса основ информатики, в том числе интегрированных банков данных, баз знаний, экспертных систем, информационно-библиографических систем и т. д.;

исследования по разработке систем с элементами искусственного интеллекта для сферы образования;

исследования по определению содержания и созданию баз знаний, экспертных систем и других интеллектуальных средств для сферы образования, производство информационных ресурсов;

исследование и разработка общесоюзной системы спутниковой связи, обеспечивающей потребности народного образования;

разработка методов оценки эффективности использования средств НИТ в образовании, в том числе методики испытания учебных программно-аппаратных продуктов с оценкой их педагогической эффективности;

разработка отраслевых нормативов по использованию средств информатики в образовании, в том числе по вопросам сохранения авторских прав разработчиков программных средств, по тиражированию, распространению и сопровождению программных средств;

анализ и обобщение международного опыта информатизации образования.

Общесоюзный центр новых информационных технологий (НИИВШ) будет проводить конкурсы, к участию в которых будут привлекать организации и институты АН СССР, АПН СССР, ГКВТИ СССР и учебные заведения всех уровней. Приемка результатов НИОКР будет осуществляться межотраслевыми экспертными комиссиями. При формировании программ предусматриваются в качестве приоритетных совместные исследования с соответствующими учебными заведениями и организациями стран — членов ЮНЕСКО.

К числу основных направлений относятся вопросы совершенствования системы подготовки, переподготовки и повышения квалификации кадров в области информатизации образования.

Совершенствование переподготовки и повышения квалификации должно строиться на основе:

перехода от «каскадного» метода подготовки преподавательского состава к двухступенчатому;

усиления практической направленности подготовки;

улучшения технической оснащенности ИПК и ФПК;

обеспечения экономической заинтересованности преподавательского состава и специалистов в повышении квалификации в области информатизации и использования полученных знаний в практической деятельности;

организации периодической переквалификации преподавательского состава с проверкой уровня подготовки в области НИТ;

повышения престижности специалистов в области информатизации, предоставление им права защиты диссертаций и присуждение им ученых степеней по НИТ;

организации обучения (стажировки) преподавателей ИПК, ФПК, учебных центров по подготовке специалистов в области информатизации за рубежом. Количество командированных для этих целей специалистов на первых этапах информатизации должно составлять порядка 3000 человек.

Информатизация образования возможна только при наличии необходимого количества и качества средств инфор-

матизации, и в первую очередь вычислительной техники. Выпускаемая отечественной промышленностью вычислительная техника и средства коммуникации обладают низкой надежностью и не удовлетворяют техническим, психолого-педагогическим и санитарно-гигиеническим требованиям, предъявляемым к средствам информатизации, используемым в учебном процессе.

При оснащении средней школы необходимо отказаться от примитивных «школьных» компьютеров, обучение школьников должно проводиться на современных ПЭВМ, обладающих возможностью диалога с обучаемым, развитой графикой, включая средства мультимедиа, звукового сопровождения программ. Более высокая стоимость таких компьютеров может быть частично компенсирована использованием этих компьютеров в вечернее время для обучения взрослых, предоставления специального времени предприятиям и т. д.

В этих условиях целесообразно:

обеспечить закупку для средних учебных заведений примерно 1 млн. персональных ЭВМ зарубежного производства;

организовать на конкурсной основе с привлечением отечественных предприятий, совместных предприятий и зарубежных фирм производство и поставку ПЭВМ, отвечающих требованиям учебного процесса;

организовать на базе высших и средних учебных заведений совместные учебно-производственные предприятия по

сборке современных ПЭВМ, совместимых с IBM PC;

обеспечить целевое распределение вычислительной техники, предоставляя ее в первую очередь центрам НИТ, организациям АПН СССР, ведущим вузам — разработчикам современного учебно-методического и программного обеспечения, ИПК системы народного образования, регионам и учебным заведениям, являющимся полигонами по экспериментальной проверке НИТ в обучении, а также институтам и организациям других министерств и ведомств, участвующим в соответствующих НИР.

Первоочередная поставка вычислительной техники организации или учебному заведению обуславливается заключением в рамках НИР по информатизации образования договора между головной организацией по НИР и данной организацией (учебным заведением) на выполнение определенного объема работ в указанные сроки с передачей результатов работ в отраслевой фонд алгоритмов и программ.

Целесообразно предусмотреть в качестве приоритетного направления комплексов работ по развертыванию общесоюзной системы спутниковой связи (как части международной системы космической связи, создаваемой под эгидой ЮНЕСКО), обеспечивающей ускоренное внедрение новых информационных технологий в обучении и развитие вузовских научных исследований на основе распределенных баз данных и знаний.

14

Уважаемые читатели!

В предыдущих номерах журнала мы опубликовали методические материалы к действующим учебникам информатики. Редакции хотелось бы знать мнение читателей, особенно учителей, о новых учебниках, о тех подходах, которые предлагают авторы. Через редакцию можно задать им вопросы, высказать замечания по поводу изложения тех или иных тем. Ждем ваших откликов.



Н. САДОВНИКОВА,
ведущий специалист Главного управления содержания образования,
методов обучения и воспитания МНО РСФСР
М. РАТИНСКИЙ,
заведующий кабинетом информатики МОИУУ

«Информатика» — базовый предмет в школах РСФСР

Школа является одним из первичных информационных учреждений. В ней изучают основы наук, необходимых человеку для дальнейшей трудовой деятельности. Именно с этих позиций, по нашему мнению, и следует рассматривать значение введенного в 1985 г. предмета «Основы информатики и вычислительной техники». Так как роль информатики в жизни современного общества возрастает, этот предмет призван стать одним из фундаментальных в системе среднего образования, а кроме того, способствовать устранению некоторых негативных явлений, имеющих место в современной школе.

Поэтому крайне важно дать диалектическое определение информатики как науки, а не сводить его, как это делается в настоящее время, к решению ряда несомненно важных задач с помощью определенного технического средства. Для правильного решения возникающих проблем необходимо провести глубокий научный анализ понятия «компьютерная грамотность», не замыкая его лишь на навыки использования ЭВМ. Необходимо учитывать, что усвоение готовых знаний без овладения деятельностью, в результате которой эти знания получены, формирует догматическое, авторитарное и в конечном счете схоластическое, метафизическое знание. Нам же нужно воспитывать у учащихся диалектическое мышление. Более того, превращение людей лишь в потребителей знаний означало бы не только потерю ими способностей к продуцированию знаний, но и обесценивание получаемой информации.

Обучение в школе должно быть неразрывно связано с формированием у учащихся необходимых элементов культуры. Информационная культура призвана помочь человеку работать в условиях различных информационных тех-

нологий. Ее проводником и должна стать информатика — наука о структуре и общих свойствах информации. Учитывая развитие современной индустрии информатики все лица после завершения обучения в общеобразовательной школе или другом среднем учебном заведении должны обладать первым уровнем компьютерной грамотности, который включает: умение достаточно уверенно пользоваться устройствами ввода информации и управления работой компьютера, а также набором готовых средств программного обеспечения для решения задач в интересующей области и быту (в первую очередь это текстовые и графические редакторы, базы данных, электронные таблицы и т. д.). Кроме того, одной из основных задач школьного курса информатики, по нашему мнению, является развитие у учащихся навыков структурно-логического мышления.

К проблеме, требующей глубокой проработки и изучения, относится анализ взаимосвязей информатики с другими предметами школьной программы. Эти взаимосвязи не должны ограничиваться лишь составлением программ для решения отдельных задач или циклов задач. Настала необходимость глубокой перестройки содержания школьных предметов, направленной не только на использование компьютеров в их преподавании, но и на подготовку учащихся в этих курсах к усвоению самого предмета «Информатика».

Именно с этих позиций необходимо оценивать «Базисный учебный план», предложенный Гособразованием СССР, и разработанные на его основе Министерством народного образования РСФСР экспериментальные учебные планы для средних общеобразовательных школ.

Анализируя «Базисный учебный план»

Гособразования СССР в двух его вариантах, относящихся к республиканским компонентам, можно встретить упоминание предмета, а точнее, термина «информатика» в качестве основ, интегрированных с предметом «Математика» в старших классах. Что это: непонимание основных задач современности — ведь во всем мире идет движение к информационному обществу — или желание спрятать просчеты в первоначальной идеологии становления этого предмета в школе, т. е. подмену овладения основами науки ее технологией?

Действительно, нет отдельного предмета, нет и ответственных за навязывание непродуманной и нежизненной концепции его преподавания. А школьная математика? Она уже стерпела и терпит различные новации, авось стерпит и эту!

Принципиальных возражений против предложения Гособразования СССР о введении в курс школьного обучения интегрированного предмета «Математика и информатика» нет. Но, на наш взгляд, этот предмет должен относиться к союзно-республиканской компоненте и преподаваться в V—IX классах, в которых и математика преподается диффе-

ренцированно. Объем часов непосредственно на раздел «Информатика» должен составлять около 64 ч. В этом случае в учебные планы Гособразования СССР органически вписываются экспериментальные учебные планы Министерства народного образования РСФСР, в которых дифференцированно, в зависимости от направления подготовки учащихся старших классов, дополнительно вводится самостоятельный предмет «Информатика» в объеме 64—102 ч. Они направлены в каждую территорию РСФСР, а также опубликованы в информационном сборнике МНО РСФСР № 18 за 1990 г. По нашему мнению, это позволит реально решить вопросы, связанные с практическим внедрением в процесс обучения современных информационных технологий.

В массовой школе закладываются основные знания. Поэтому столь важная в настоящее время социальная задача, как формирование первоначальных навыков информационной культуры, должна решаться именно в школе. От этого зависит успех внедрения современных технологий на всех уровнях народного хозяйства, а следовательно, и его прогресс.

УЧЕБНЫЙ КЛАСС ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ УКВТ-89

Производственный кооператив «КОМПЬЮТЕР»

производит поставку и установку компьютерного класса на базе ПК-01 «Львов» и видеомониторов типа «Электроника 328ТЦ» с количеством рабочих мест от 1 до 25. Класс оборудован локальной сетью типа «звезда». В комплект поставки входит математическое обеспечение, включающее игровые и учебные программы.

Ориентировочная цена УКВТ-89 из 11 рабочих мест составляет 28—40 тыс. рублей.

Срок поставки — 1 месяц после получения аванса.

Возможна поставка класса на базе компьютера, программно совместимого с компьютером «Спектрум».

Телефоны: 693-819 — с 9 до 17 ч,
644-151 — после 20 ч.

Адрес: 290031, г. Львов, аб. я. 9143

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

Исполнители

Исполнитель — одно из основных понятий в курсе раннего обучения информатике. Школьники уже встречались с исполнителями в теме «Алгоритмические этюды». Дети, хотя и не пользовались специальной терминологией, познакомились с примерами исполнителей, понятием среды, реакциями на ошибки. Наступает момент, когда необходимо назвать вещи своими именами и обобщить полученные ранее знания.

Урок 11.2. Тема: «Исполнитель, система команд»

План урока:

Проверка домашнего задания.

Общее представление об исполнителе, анализ бытовых ситуаций.

Система команд исполнителя.

Аппаратные исполнители-роботы.

Обозначение (название) команды, способ передачи ее исполнителю, выполнение.

Два вида ошибок: НЕ МОГУ (семантические ошибки) и НЕ ПОНИМАЮ (синтаксические ошибки).

Игра в исполнителя.

Лабораторная работа: исполнитель Квадратик.

Домашнее задание: придумать систему команд для исполнителя.

Методический комментарий к этому уроку сильно сокращен в связи с тем, что материал учебника говорит сам за себя.

§ 4.1. Исполнитель, система команд

В сказках часто встречаются волшебные устройства, которые послушно ис-

полняют волю сказочных героев. Помните, как, выбравшись из чащи леса на полянку, Иван-царевич видит избушку на курьих ножках? Иван дает ей команду:

— ИЗБУШКА, ИЗБУШКА! ВСТАНЬ КО МНЕ ПЕРЕДОМ, К ЛЕСУ ЗАДОМ!

И волшебная избушка ее исполняет.

Команда должна быть задана очень точно, иначе нечего рассчитывать, что устройство окажется послушным. Например, сказка про Али-Бабу и сорок разбойников. Ее события связаны с волшебной дверью, которая могла по команде открываться, впуская или выпуская человека в пещеру с сокровищами. Помните, какой печальный конец нашел в пещере жадный Касым, забывший эту команду?

И волшебная дверь, и избушка на курьих ножках ведут себя одинаково:



OldPC.su

7 0 0 3

музей компьютеров

они умеют понимать и очень точно исполнять некоторые команды. Поэтому их можно назвать исполнителями.

Исполнители знакомы вам не только по сказкам. Дрессированная собака — исполнитель: она понимает и исполняет те команды, которым научил ее хозяин. Мальчика, который по просьбе мамы моет на кухне посуду, тоже можно назвать исполнителем. Исполнители много раз встречались вам на уроках информатики. Перевозчик был одним из первых исполнителей, с которым вы познакомились. Переливашка, Конюх, Ханойская башня — все это исполнители.

Исполнителем называется человек, коллектив, животное или техническое устройство, которые понимают и умеют точно исполнять задаваемые им команды.

18

Набор понятных исполнителю команд называется системой команд этого исполнителя, или, сокращенно, СКИ.

Например, у избушки на курьих ножках СКИ составлена из одной-единственной команды: «ПОВЕРНИСЬ КО МНЕ ПЕРЕДОМ, К ЛЕСУ ЗАДОМ!»

Вопросы и упражнения

1. Система команд исполнителя Муравей состоит из команд ВВЕРХ, ВНИЗ, ВЛЕВО, ВПРАВО. Понятны ли Муравью команды КВЕРХУ, НАПРАВО и ВЫМОЙ ЛАПЫ? Почему?

2. Приведите примеры исполнителя-животного, исполнителя-человека, исполнителя-коллектива, исполнителя-машины.

3. Является ли исполнителем компьютер?

4. Чем отличается исполнитель-человек от исполнителя-машины?

§ 4.2. Что надо знать об исполнителе?

Катя Пушкиова, Петя Кук и Вова Абрамов торопились на урок информатики.

— Сегодня учитель обещал познакомиться нас с новым исполнителем. — сказала Катя мальчишкам.

— Это значит, — ответил Петя, — что он нам расскажет новую СКИ.

— Знать название команд мало, — вмешался в разговор Вова. — Ведь кроме названия надо знать еще, во-первых, как каждую из этих команд можно пере-

дать исполнителю и, во-вторых, как исполнитель выполнит эту команду.

— Я согласна, — поддержала беседу Катя. — В самом деле, разные исполнители получают свои команды по-разному. Например, Мудрый Кролик получает команды ВЛЕВО, ВПРАВО, ВВЕРХ, ВНИЗ, когда человек, управляющий Кроликом, нажимает соответствующие клавиши со стрелками.

— И конечно, важно знать, как выполняется команда, — согласился Петя. — Я должен точно знать, что делает Кролик по команде ВВЕРХ: поднимает вверх правую лапу или перемещается вверх на одну клетку.

Итак, знакомясь с исполнителем, необходимо выяснить:

1. Какие команды входят в СКИ.
2. Как команды передаются исполнителю.
3. Как исполнитель выполняет команды.

§ 4.3. Сообщения исполнителя НЕ ПОНИМАЮ и НЕ МОГУ

Петя Кук остановился на полпути и не знает, куда дальше идти — вперед или назад. Попробуйте ему дать записочку со словом ВПИРЕТ. Он, конечно, улыбнется, глядя на ваши ошибки, но все-таки поймет, что надо делать.

А вот если в СКИ исполнителя-робота имеется команда ВПЕРЕД, а вы наберете это слово с ошибкой, робот напишет на экране:

НЕ ПОНИМАЮ

Так всегда отвечают роботы, когда им дают команду, не входящую в их СКИ. Робот принимает команды к исполнению только тогда, когда они отданы буквально так, как записано в СКИ.

Можно задать исполнителю команду, имеющуюся в СКИ, и тем не менее поставить его в затруднительное положение. Представьте себе, что коза находится на правом берегу, а вы отдаете Перевозчику команду КОЗУ НА ПРАВЫЙ БЕРЕГ. Исполнитель напишет на экране:

НЕ МОГУ



Так всегда отвечают роботы, когда не могут по каким-либо причинам выполнить команду из своей СКИ.

НЕ ПОНИМАЮ все исполнители говорят в одном и том же случае: когда им задают команду, не входящую в СКИ.

НЕ МОГУ у каждого исполнителя свое: Перевозчик не может везти козу, если ее нет на одном с ним берегу; Мудрый Кролик откажется выполнять команду **ВПРАВО**, уперевшись в правую стенку лабиринта. Значит, знакомясь с исполнителем, надо знать, когда исполнитель говорит **НЕ МОГУ**.

Вопросы и упражнения

1. Исполнитель Перевозчик находится на правом берегу вместе с козой, а волк с капустой — на левом. В таком положении Перевозчику приказано **КАПУСТУ НА ЛЕВЫЙ БЕРЕГ**. Как он отзовется на такое поручение?

Сравнивая ситуацию **НЕ ПОНИМАЮ** и **НЕ МОГУ**, учитель подчеркивает, что первая из них — общая для всех исполнителей: команда не принадлежит СКИ. А вот ситуации **НЕ МОГУ** специфичны не только для каждого исполнителя, но и для каждой команды. Поэтому, рассказывая о ситуациях, в которых исполнитель выдает сообщение **НЕ МОГУ**, надо отдельно рассматривать каждую команду.

Важно, чтобы дети поняли, что ошибки совершает не робот, а человек, который им управляет.

Итак, понятие исполнителя определено. Сейчас уместно придумать исполнителя, чтобы в него поиграть. Учитель вызывает одного из учеников к доске, чтобы тот сыграл роль исполнителя. Вместе с учителем класс создает и записывает его СКИ. Конечно, предложения должны поступать от учеников,

но дети пока еще не придумывали исполнителей, поэтому фантазия ребят непредсказуема. Учителю важно организовать поток предложений, призванных составить систему команд, и, возможно, подсказать несколько команд.

СКИ может быть, например, такой:
название команды выполняемые действия

| | |
|---------------------|-------------------------------|
| ПОДНИМИ РУКУ | поднять одну руку |
| ОПУСТИ РУКУ | опустить одну руку |
| СЯДЬ | сесть на стул |
| ВСТАНЬ | встать со стула |
| ПРЫГАЙ | три раза подпрыгнуть на месте |
| МЯУКАЙ | произнести «мяу» |

Зафиксировав СКИ, учитель спрашивает у детей название или обозначение каждой команды; каким образом они передаются исполнителю (голосом, запиской, жестами ...); как выполняется каждая команда.

Затем дети по очереди предлагают исполнителю выполнить одну из команд СКИ. Ученик-исполнитель должен либо выполнить полученную команду, либо произнести **НЕ ПОНИМАЮ** (когда полученная команда не входит в СКИ, например **ПРЫГНИ** вместо **ПРЫГАЙ**) или **НЕ МОГУ** (когда команда не может быть выполнена, например две руки уже подняты, а поступает команда **ПОДНИМИ РУКУ**).

Возможно, что игра пройдет весело, шумно. Это оправдано, если в игре дети усвоят понятие СКИ. Если дети поддержат предложенный учителем высокий темп игры, можно будет удовлетворить желания еще одного-двух школьников, стремящихся показать себя в роли исполнителей. Во избежание потерь времени на этом очень плотном уроке повторную игру лучше провести, не меняя СКИ.

§ 4.4. Как познакомиться с исполнителем?

Когда человек хочет работать с новым исполнителем, он должен выяснить:

1. В какой среде (обстановке) работает исполнитель и какую работу он может выполнять?
2. Какая СКИ у исполнителя, а именно:
 - 2.1. Какие команды входят в СКИ?

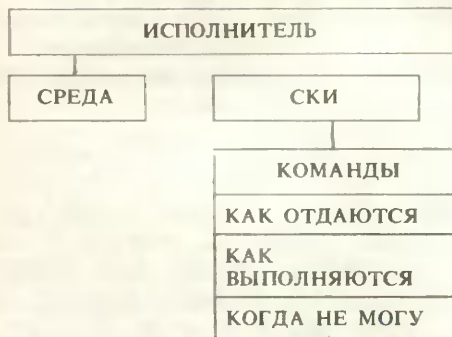
2.2. Как команды передаются исполнителю?

2.3. Как команды выполняются исполнителем?

2.4. Когда выполнение команды приводит к ситуации НЕ МОГУ?

Ниже приводится схема знакомства. Используйте ее при выполнении упражнений.

Схема знакомства с исполнителем



Пример. Петя Кук обучает младшего брата работе с Мудрым Кроликом.

— Посмотри, Вася, на схему знакомства. Сначала я расскажу тебе *среду* исполнителя: Кролик работает в лабиринте. Кроме Кролика в лабиринте разбросаны мешки с капустой. Точками отмечен склад. Кролику надо собрать все мешки в лабиринте и перенести их на склад.

Теперь, Вася, давай рассмотрим *СКИ* Мудрого Кролика. Вот какие у него команды: ВВЕРХ, ВНИЗ, ВЛЕВО, ВПРАВО.

Эти команды *отдаются* кролику нажатием на клавиши-стрелки.

Исполняя эти команды, Кролик перемещается в соответствующем направлении на один шаг. Если на пути встречается мешок с капустой, он толкает его перед собой.

Кролик *не может* выполнить команду, если он упирается в стенку лабиринта, не может он также толкать перед собой два мешка или больше.

§ 4.5. Лабораторная работа: Квадратик

Квадратик умеет рисовать на экране



квадрат, треугольник, круг, может закрасить нарисованные фигуры.

СКИ Квадратика

КВАДРАТ
ТРЕУГОЛЬНИК
КРУГ
ЗАКРАСКА
ОЧИСТКА

Квадратик *не может* рисовать фигуру, если на экране уже есть рисунок.

Задания.

1. Нарисуйте квадрат.
2. Нарисуйте треугольник.
3. Нарисуйте окружность.
4. Напишите команды, которые нарисуют в той же последовательности те же фигуры, закрашивая их.

Особенностью этой работы является то, что учитель не дает предварительных объяснений. Дети должны по возможности самостоятельно изучить исполнителя по описанию лабораторной работы, попробовав его реакции на все возможные действия.

Урок 11.3. Тема: «Исполнители, схема знакомства»

План урока:

Проверка домашнего задания.

Игра в исполнителя.

Что надо знать про нового исполнителя, чтобы начать с ним работать.

Анализ исполнителя Квадратик по общей схеме знакомства с исполнителем.

Практическая работа: Курсор. Особенности Курсора с точки зрения общей схемы знакомства.

Этот урок посвящается главным образом закреплению понятия исполнителя и элементов схемы знакомства с исполнителем. С Квадратиком — очень простым исполните-

лем — к детям приходит осознание важнейших понятий темы. Однако искусственность Квадратика понятна даже детям. А вот исполнитель Машинист, появляющийся на следующем уроке, вводит школьников в мир моделей реальных явлений, не отменяя, впрочем, ни игру, ни компьютерные упражнения.

Урок 11.4. Те м а: «Исполнители. Машинист»

П л а н у р о к а:

Проверка домашнего задания:

опрос по схеме знакомства с исполнителем; игра в исполнителя.

Обсуждение исполнителя Машинист (задачи о сортировочной площадке как модели реальной ситуации).

Простая задача сортировки состава.

Обсуждение аварийных ситуаций.

Домашнее задание: записать алгоритм перестановки первого вагона на нижний путь.

Вступительная часть урока завершается уже знакомой игрой в созданный детьми исполнитель. На этот раз создатель исполнителя должен представить его классу по схеме знакомства.

Когда игра закончена, учитель может вступить в беседу об одном принципиальном отличии формального исполнителя-робота от неформального, обладающего волей:

— Дома за завтраком вы иногда говорите маме, угощающей вас манной кашей: «Не хочу». Если какой-нибудь хулиган предложит вам стрельнуть из рогатки в окно, вы, не долго размышляя, откажетесь это делать. Таким образом, человек может и отказаться выполнить команду, если она ему не нравится, исправный робот — никогда. Это одно из самых важных свойств исполнителя, одно из главных отличий человека от робота.

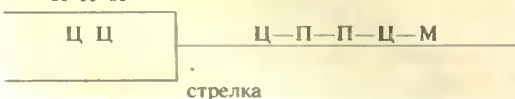
(Отсюда, вообще говоря, следуют важные нравственные выводы: за действия исполнителя-робота отвечает управляющий им человек; когда исполнителем является человек, он сам отвечает за свои действия. Младшие школьники должны понимать это, но прямая дискуссия на эту тему более уместна в старших классах.)

§ 4.6. На железнодорожной станции

Ребята из класса Пети Кука давно мечтали сходить на железнодорожную станцию. И вот учитель повел их на обещанную экскурсию.

Больше всего Куку запомнился небольшой маневровый тепловоз, который беспрестанно ездил взад и вперед, таская за собой вагоны. Петя буквально замер на месте, наблюдая, как на сортировочной горке машинист тепловоза из одного длинного товарного поезда составлял два других, поменьше, в каждом из которых оказывались вагоны одного вида: в одном платформы, а в другом — цистерны. Машинист оттащивал назад сборный поезд, а затем снова возвращался к стрелке. Стрелку надо было перевести в зависимости от того, какой вагон оказывался хвостовым. Тогда тепловоз толкал последний вагон на нужный путь и отцеплял его. Потом все начиналось сначала.

П П П



21

Приглядевшись внимательно, Петя вдруг обнаружил, что в кабине машиниста никого нет. От удивления Петя даже вскрикнул. Но учитель все объяснил:

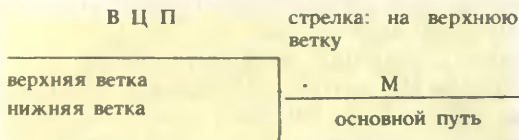
— Машиниста-человека в кабине действительно нет. Зато есть робот. Это Машинист-исполнитель. Он выполняет команды дежурного по станции, которые получает по радио.

§ 4.7. Исполнитель Машинист

Посмотрите на рисунок. Среда Машиниста включает три железнодорожных пути со стрелкой и железнодорожный состав, состоящий из тепловоза (М) и трех вагонов: крытого (В), цистер-



ны (Ц), платформы (П). Исполнитель Машинист, управляемый человеком, сортирует вагоны в нужном порядке — такая у него работа на станции.



КОМАНДЫ

СКИ Машиниста

ВПЕРЕД
 НАЗАД
 СТРЕЛКА
 ПРИЦЕПИ
 ОТЦЕПИ

22

КАК ОТДАЮТСЯ

Команды исполнителю набираются на клавиатуре компьютера заглавными русскими буквами точно так, как они обозначены в СКИ.

КАК ВЫПОЛНЯЮТСЯ

ВПЕРЕД — если тепловоз стоит в начале основного пути, то по этой команде он продвигается до стрелки; если же он стоит у стрелки, то команда заставляет тепловоз идти в тупик по той из веток, путь на которую открыт стрелкой.

НАЗАД — по этой команде тепловоз движется задним ходом и откатывается в начало основного пути.

СТРЕЛКА — эта команда переводит стрелку на развилке из двух путей; если первый путь был закрыт (и значит, открыт второй), то команда **СТРЕЛКА** открывает первый путь и закрывает второй.

ПРИЦЕПИ — по этой команде к тепловозу (или к целому составу, который везет тепловоз) прицепляется вагон (внимание: только один!)

ОТЦЕПИ — команда отцепляет последний вагон состава (только один вагон; чтобы отцепить два вагона, надо подать две команды **ОТЦЕПИ**).

КОГДА НЕ МОГУ

| Команда | Когда возникает ситуация НЕ МОГУ |
|-----------------|--|
| ВПЕРЕД НАЗАД | Тепловоз (или весь состав) упирается в тупик |
| СТРЕЛКА | Тепловоз не у стрелки |

| | |
|---------|--|
| ПРИЦЕПИ | Тепловоз не у вагонов Прицеплены все вагоны |
| ОТЦЕПИ | Состав не в тупике Отцеплены все вагоны |

Вопросы и упражнения

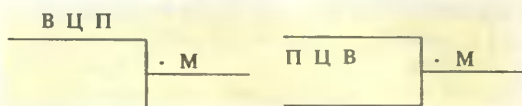
1. Какую информацию обрабатывает Машинист?

2. Разберитесь в описании приведенного в этом параграфе алгоритма. Для этого нарисуйте на листе бумаги пути, представьте вагоны и тепловоз спичечными коробками, а стрелку сделайте из спичек. Представьте себе, что вы и есть исполнитель Машинист, и выполните все команды этого алгоритма.

3. Пользуясь командами Машиниста, составьте алгоритм, по которому вагоны, стоящие на верхней ветке, окажутся на нижней, но в обратном порядке (начальное и конечное положение показаны на рисунках).

Дано:

Надо:



После обсуждения СКИ Машиниста предлагается составить алгоритм, по которому Машинист переставил бы платформу на нижний путь.

Дано:

Надо:



Восприятие Машиниста, вообще говоря, неоднозначно: одним он представляется роботом, разместившимся в кабине локомотива; другие могут подумать, что исполнитель бежит взад и вперед по сортировочной площадке, переводя стрелку и сцепляя вагоны. Для последующих бесед о Машинисте и его задачах можно принять соглашение: Машинист — робот, который находится на локомотиве. Он получает команды по радио от дежурного по станции, составившего очередной алгоритм сортировки состава. Для выполнения команд Машинист располагает автоматическими устройствами, которые позволяют ему не выходя переводить стрелку, сцеплять и расцеплять вагоны. Однако надо дать понять детям, что возможны и другие варианты.

В учебнике введены обозначения для вагонов и локомотива. При объяснении на уроке эти обозначения полезно заменить более выразительными объектами. Например, среда исполнителя воплощена на магнитной доске; вагоны и локомотив — соответствующие изображения, вырезанные из бумаги и наклеенные на магниты.

При решении первой задачи важно не просто выписать последовательность команд, а изобразить состояния среды, возникающие после выполнения каждой команды (в журнальном варианте эти рисунки-схемы опущены). Поэтому учитель, выполнив очередной шаг перемещения «вагонов» на магнитной доске, фиксирует этот шаг на кодоскопе. Взяв на себя фиксирование истории выполнения, учитель может вызвать к доске ученика, который будет, следуя получаемым командам, перемещать «вагоны» на магнитной доске. Надо обратить внимание детей на то, что ученик сейчас играет роль исполнителя Машиниста.

Комментарии делаются устно:

ВПЕРЕД (продвижение локомотива до стрелки).

ВПЕРЕД (продвижение локомотива от стрелки до состава).

ПРИЦЕПИ (присоединение ближайшего вагона к локомотиву).

НАЗАД (локомотив с вагоном отъезжает на основной путь).

ВПЕРЕД (продвижение состава до стрелки).

СТРЕЛКА (стрелка открывает путь на нижний тупик).

ВПЕРЕД (продвижение состава до нижнего тупика).

ОТЦЕПИ (вагон остается в нижнем тупике).

НАЗАД (возврат локомотива в исходную позицию).

Затем учитель переводит класс из одной формы представления среды (на магнитной доске) в другую (на экране ПЭВМ). Клавиатурные навыки детей еще далеки от совершенства, поэтому для демонстрации Машиниста выбрана столь простая задача. Тем не менее даже для этой задачи надо изобразить начальную и конечную ситуации — «Дано» и «Надо».

Дети должны самостоятельно набрать команды на своих машинах. Если класс располагает демонстрационным экраном, учитель может показывать правильные команды.

Однако учительская демонстрация не должна опережать действия учеников. Дети ориентируются на большой экран как на средство проверки, но не подсказки.

В этом простом примере задействованы все команды СКИ. Поэтому, завершив его, полезно обсудить пункт схемы знакомства «Как выполняются команды». По каждой команде можно заслушать выступление отдельного ученика.

В ходе выполнения алгоритма неизбежны ситуации НЕ МОГУ и НЕ ПОНИМАЮ. Их обсуждение — элемент урока.

Следующий урок — это еще одна встреча с Машинистом. Задачи усложняются. На этих задачах впервые появляются элементы технологии проектирования «сверху вниз»: сначала запись в словесных выражениях укрупненных этапов алгоритма, а затем — развертывание этих этапов командами из СКИ Машиниста. Не менее важный элемент урока — роль программных исполнителей как компьютерных моделей реальных объектов и процессов.

23

Урок 11.5. Тема: «Исполнитель Машинист, алгоритм, программа, модель»

План урока:

Повторение: система команд исполнителя Машинист. Проверка домашнего задания. Программа, соотношение между алгоритмом и программой.

Отношения между создателем (разработчиком) и исполнителем алгоритма.

Исполнители как модели.

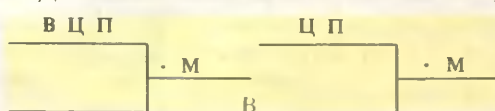
Решение задачи на перестановку состава (безмашинное).

Лабораторная работа: Машинист.

Домашнее задание: алгоритм перестановки состава.

Проведя вводную часть урока, учитель подходит к решению домашнего задания. Первое, с чего начинается решение задачи, — определить начальную информацию («Дано») и требуемый результат («Надо»). Рекомендуется записывать условия задачи в виде схем, подобных тем, что введены в § 4.7:

Дано: Надо:



Сначала надо обсудить общий план решения задачи (см. табл. 1).



24

Далее надо коллективно восстановить определение: план решения задачи называется алгоритмом. Значит, табл. 1 содержит алгоритм решения задачи.

Теперь можно приступить к детализации плана — составлению программы. Именно детальный план был содержанием домашнего задания. Однако сейчас речь идет не о простом переписывании команд из домашней тетради, а об установлении соответствия между пунктами общего плана решения — алгоритма, описанного в понятной человеку форме табл. 1 и командами из СКИ.

Включаются машины. В работе участвуют все — и те, кто сделал домашнюю работу, и те, для кого она оказалась сложной. Результат обсуждения представляется в виде табл. 2.

Таблица 2

| | |
|--|--|
| 1) Переставить состав на нижний путь | ВПЕРЕД ВПЕРЕД ПРИЦЕПИ ПРИЦЕПИ НАЗАД ВПЕРЕД СТРЕЛКА ВПЕРЕД |
| 2) Переставить пару «цистерна — платформа» на верхний путь | ОТЦЕПИ НАЗАД ВПЕРЕД СТРЕЛКА ВПЕРЕД |
| 3) Вернуть локомотив на место | ОТЦЕПИ ОТЦЕПИ НАЗАД |

Важный элемент урока — сравнение табл. 1 и 2. Табл. 1 представляет собой запись алгоритма решения задачи. А что такое табл. 2? Это другая запись алгоритма решения той же задачи.

Запись алгоритма командами СКИ (на языке, понятном исполнителю) называется программой для этого исполнителя.

Табл. 1 понятна человеку и совсем не понятна исполнителю Машинист: ведь ни одной команды из приведенных в этой таблице нет в его СКИ. Табл. 2 хорошо понятна исполнителю, но для человека трудна. При решении задач нужно всегда начинать с записи алгоритма в наиболее понятном человеку виде и только потом, когда такая запись составлена, начинать переводить ее в программу на языке СКИ.

Далее на уроке обсуждаются отношения между создателем (разработчиком) и исполнителем алгоритма. Труд исполнителя механический: он выполняет понятную ему команду или сообщает о ее невыполнимости. Тому, кто выполняет алгоритм, достаточно знать только СКИ. Труд составителя алгоритма творческий: при составлении алгоритма надо придумать способ решения задачи. Тому, кто создает алгоритм, знаний СКИ недостаточно, он должен хорошо знать предметную область и уметь мыслить. Видно, что требования к составителю и исполнителю алгоритма сильно различаются.

Вот почему составителями алгоритмов почти всегда бывают люди, а исполняют алгоритмы послушные людям роботы. Слово «почти» в последней фразе позволяет не упоминать в этом простом анализе те задачи, в которых компьютер создает алгоритмы, исполняемые впоследствии другими техническими устройствами или даже людьми: из общеобразовательного курса информатики школьнику лучше вынести «классическое» отношение между творцом-человеком и исполнителем-машиной.

Обязательная часть обсуждения отношений между человеком и исполнителем — реакция исполнителя на ошибки. Еще раз с помощью школьников рассматриваются условия, в которых возникают сообщения о синтаксических и семантических ошибках при управлении Машинистом, приводятся примеры. Сам факт возможности ошибок связывается с тем, что управление исполнителем осуществляет человек. Человеку свойственно ошибаться, исправный исполнитель рабо-

тает безошибочно. Учитель должен убедить школьников в этом важном тезисе. Убежденность в первичности человеческих и вторичности машинных ошибок поможет им впоследствии проводить методологически важную работу по проверке и тестированию алгоритмов.

Обсуждение исполнителей как моделей можно провести в соответствии с § 4.8 учебника.

§ 4.8. Модели на компьютере

Когда создается новый самолет, конструктор продумывает разные варианты будущей машины. Эти варианты могут отличаться размахом крыльев, расположением двигателей, количеством пассажирских кресел. Однако невозможно каждый раз сооружать самолет, проверяя новый вариант конструктора: это очень долго, сложно и дорого.

Поэтому прежде чем построить самолет, делают его модель или даже несколько моделей. Модель занимает немного места, изготавливается быстро, стоит она недорого. Даже если модель испортится, потеря невелика. А польза от модели огромна: испытывая ее, можно в короткие сроки ответить на многие сомнения конструктора.

Модели сооружают не только конструкторы самолетов. Модель нужна строителям. Готовят модели костюмеры, прежде чем пошить большое количество новых платьев.

Часто приходится делать модели действий, а не модели вещей. Агроном, планирующий сев, хотел бы знать, под какие растения следует отвести поля в разных уголках своего совхоза. Хорошо было бы попробовать несколько разных вариантов. Но ведь чтобы увидеть результат, надо ждать целый год, пока вырастут плоды. И вот тут на помощь приходит компьютер. Вместо того чтобы сажать семена в поле, делают модель роста растений и наблюдают на экране компьютера за созреванием урожая. Компьютеру надо давать информацию

о почве, на которой растут растения, об удобрениях, которыми подкармливают посевы, о сельскохозяйственных машинах, используемых при посеве или сборе урожая. Компьютер, обработав эту информацию, показывает на экране, как мог бы расти урожай на самом деле. Это модель работы совхоза.

Компьютерная модель может показать изображаемые действия (посев, прорастание семян, всходы растений, подкормку, прополку, сбор урожая) с любой скоростью — как при ускоренной киносъемке. Тогда год жизни совхозного поля можно увидеть на экране ЭВМ за три минуты.

Компьютеры часто используют для создания моделей. Вы сами это видели на уроках информатики. Действительно, если бы задачу о перевозке через реку надо было решать не на экране компьютера, а находясь в одной лодке с настоящим живым волком, то далеко не все дети захотели бы выступить в роли исполнителя. С помощью компьютера была создана модель Перевозчика.

Точно так же исполнитель Машинист — это модель сортировочной площадки на железнодорожной станции. Эта модель позволяет проверить, как будет работать на станции настоящий робот-тепловоз, а также научить людей пользоваться сложными техническими устройствами.

Вопросы и упражнения

1. Для каких целей люди используют модели вместо сложных машин?
2. Чем отличается модель от моделируемого устройства? Что у них общего?
3. Чем полезны исполнители, моделирующие действия? Приведите примеры.

Здесь описано лишь начало темы «Исполнители» в курсе Роботландии. В продолжении ее следуют рассказы об арифметических исполнителях, которые приводят в конце концов к модели стекового калькулятора. С его помощью школьники видят, как реально можно применить их знания по информатике на уроках математики.

Опыт обучения работе с текстовым редактором

26

Практически каждому пользователю ЭВМ приходится в той или иной мере прибегать к системам редактирования текстов. Поэтому естественно включить навыки работы с текстовыми редакторами в число ключевых компонентов компьютерной грамотности. Возможности наиболее совершенных текстовых редакторов чрезвычайно разнообразны. Однако учащимся нет необходимости осваивать все функциональные возможности. Основное требование — практическое знание базовых команд, которое дает «систему отсчета» для освоения любого текстового редактора. К числу этих команд относятся: обращение к системе за помощью, вставка и удаление символов и строк (в том числе пустых), копирование участков текста, поиск по образцу.

Сопоставительный анализ четырех моделей текстовых редакторов

Для экспериментального обучения (на КУВТ «Ямаха») были использованы четыре модели текстовых редакторов. На первом этапе проводилось обучение работе с редакторами «Микромир», «лосьн» и SCED. Задачей этапа была оценка возможностей редакторов и ограничений их применения в учебном процессе.

Методика обучения была достаточно простой. Ученикам объяснялись элементарные правила работы с редактором и предлагалось напечатать небольшой текст. Затем на этом тексте демонстрировались и отрабатывались функциональные возможности редактора. На заключительном, контрольном занятии учащиеся должны были перепечатать некоторый текст и самостоятельно произвести с ним заданные действия.

Анализ результатов позволяет сделать выводы по каждому из редакторов.

«Микромир» обладает большими функциональными возможностями, относится к классу профессиональных текстовых редакторов, не требует зна-

ния английских слов и имеет чрезвычайно полезную возможность — «откатку», с помощью которой пользователь может «прокрутить» свои действия в обратной последовательности, восстановив то состояние, которое было в прошлом. Это особенно важно при обучении: учащийся может не бояться необратимых изменений, любая ошибка легко исправляется, и это стимулирует его к активному изучению возможностей редактора.

Однако «Микромир» имеет ряд особенностей, препятствующих его широкому использованию в учебном процессе средней школы.

Во-первых, на экране представлена 80-символьная строка, а не 40-символьная, в результате буквы становятся трудно различимыми.

Во-вторых, функциональная нагрузка на клавиши у «Микромира» очень велика и при случайном нажатии могут быть получены самые неожиданные результаты. Хотя возможность «откатки» значительно облегчает положение учащегося, побочные эффекты отвлекают его от решения учебных задач.

В-третьих, ввод команд производится с помощью сложных сочетаний функциональных клавиш. Разработчики сознательно свели к минимуму «внешние подпорки» пользователя, добиваясь непосредственности преобразований. При этом резко возросла сложность освоения навыков ввода команд. Как правило, учащиеся между занятиями успевали забыть сочетания клавиш, соответствующие освоенным командам, и в начале урока вынуждены были восстанавливать навыки ввода по «шпаргалке».

В-четвертых, система недостаточно «прозрачна» для начинающего пользователя, так как на экране не индицируется режим (вставки или замены, работы с текстом или определения макрокоманды).

Вывод: интерфейс пользователя редактора «Микромир» рассчитан скорее на профессионала, чем на новичка. SCED

(SCreen EEditor) предназначен в основном для написания и редактирования текстов программ. Работа с этим редактором требует специальных знаний, в том числе английских названий, форматов и способов употребления команд. Принципы работы редактора SCED не позволяют пользователю эффективно применять знания, полученные в процессе «бескомпьютерной» работы с текстом. В этом смысле SCED существенно отличается от «Микромира», в котором аналогия текста со «свитком» помогает пользователю ориентироваться в процессе работы.

«Лосьн» специально предназначен для начальных этапов обучения созданию и преобразованию текстов. Команды редактирования выполняются при нажатии функциональных клавиш, причем на экране (в нижней строке) приводятся названия соответствующих команд. Таким образом, все возможные варианты действий находятся в поле зрения обучаемого. Кроме того, в редакторе «лосьн» предусмотрена более подробная подсказка обучаемому. При нажатии на клавишу «ПОМОГИ» система переходит в особый режим — нажатие на любую функциональную клавишу вызывает появление на экране текста, объясняющего использование команды, соответствующей этой клавише.

К недостаткам редактора «лосьн» можно отнести его невысокую техническую надежность (в течение урока одна-две машины в результате ошибок программы переходили в командный режим Бейсика, при этом набранный текст уничтожался). Кроме того, практика показала, что ученики очень плохо воспринимают с экрана ЭВМ чисто текстовые инструкции. Возможно, использование графики сделало бы систему помощи редактора «лосьн» более эффективной.

На втором этапе проводилось обучение работе с текстовым редактором СТР. Основной задачей этапа была апробация методики обучения работе с текстовым редактором, о которой будет сказано ниже. Относительно возможностей этого редактора можно сделать следующие выводы.

СТР (Сетевой Текстовый Редактор)

входит в пакет «Информатика-87», созданный НИИ ВТ АПН СССР. Это упрощенная версия редактора TOP, предназначенная для работы с сетью. К достоинствам редактора относятся встроенная «шпаргалка» команд, к которой можно обратиться в любой момент, возможность смены 40-символьной строки на 80-символьную и обратно, удобная пересылка редактора и текстовых файлов по сети.

В процессе обучения выявились и недостатки редактора (вернее, технические недоработки его реализации):

некоторые ученические машины после пересылки, команд копирования, поиска по образцу и разрезания строки функционируют неправильно;

если строка превышает 80 символов, остаток не переносится на другую строку, а уничтожается;

при нажатии на некоторые клавиши происходят неожиданные действия (например, если ученик при нажатии на CTRL затрагивает соседнюю клавишу TAB, то вставляется пустая строка);

клавишам команд соответствуют первые буквы не русских, а английских названий, что затрудняет их запоминание учащимися.

Таким образом, ни один из рассмотренных редакторов не отвечает в полной мере требованиям учебного процесса школы и ПТУ. Оценивая возможности доработки перечисленных редакторов, заметим, что наилучший путь состоит в создании упрощенной версии «Микромира» (с 40-символьной строкой, более удобным вводом команд и ограниченными функциональными возможностями).

Апробация методики обучения работе с текстовым редактором

На втором этапе проводилось обучение четырех групп учащихся работе с текстовым редактором СТР. Методика обучения состояла в следующем.

Первое занятие во всех четырех группах проводилось по одной и той же схеме: учащихся знакомили с понятием текстового редактора, особенностями «электронных» текстов. На этом же занятии проводилась оценка уровня сфор-

мированности клавиатурных навыков.

Второе и третье занятия проводились по разным схемам. Первым двум группам учащихся последовательно демонстрировались (с требованием повторить на своих ПЭВМ) основные команды редактора СТР. На следующем уроке имитировалась ситуация деловой переписки: по локальной сети на рабочие места учащихся рассылался текст официального письма; учащимся предлагалось представить себя на месте директора завода стройдеталей, который должен составить и отправить аналогичный запрос смежникам. Для этого требовалось скопировать текст «входящего» документа и внести в него нужные коррективы.

28

В двух других группах порядок работы был иным. На втором занятии сразу же начиналась работа с письмом. В процессе работы учащимся показывались (без подробного объяснения) основные действия. На третьем занятии объяснялась вся совокупность команд редактора, причем в качестве системы отсчета активно использовался уже имеющийся опыт работы по подготовке письма.

Схема четвертого занятия во всех группах была одинаковой. Учащиеся должны были напечатать стихотворение, в котором имелся рефрен, повторяющийся с небольшими изменениями. Требовалось применить команду копирования, отредактировать эти копии, изменить пространственное расположение отдельных частей текста (увеличить число пробелов между строфами, сдвинуть все строки рефрена на три позиции вправо и т. д.).

На пятом занятии проводилась контрольная работа. Учащимся были розданы несколько вариантов однотипного задания. Во всех вариантах требовалось напечатать исходный текст и внести в него изменения.

На основе результатов выполнения контрольных заданий и индивидуально-го опроса учащихся оценивались их умения по следующим показателям: печать текста, вызов подсказки, вставка и удаление отдельных символов (строк), разрезание и склейка строк, копирование, поиск по образцу.

Анализ результатов экспериментального обучения работе с редактором СТР

На последнем занятии было проведено «экспресс-анкетирование» учащихся. Им было предложено сформулировать свои трудности при обучении работе с редактором.

Анализ результатов выполнения контрольного задания и «экспресс-анкетирования» позволил сделать следующие выводы.

У большинства учащихся возникли трудности, связанные с копированием текста. Многие сформулировали эти трудности словами «плохо запоминаю, как...», значительная часть указала, что особенно сложным является определение места, в которое должны быть вставлены новые строки.

Анализ клавиатурных навыков учащихся, проведенный с помощью программы «КЛАВИАТУРА», дал следующие результаты. Среднее время печати символа у большинства учащихся находилось в пределах 0,6—1,4 с, время печати символа в ключевых словах Бейсика было значительно ниже. Для большинства учащихся оно составляло 0,3—1,2 с.

Правильно напечатать заданный текст смогли все учащиеся, обратиться к подсказке — около 70 %, выполнить вставку и удаление символов и строк — около 50 %, копирование группы строк — около 40 %, поиск по образцу — около 20 %.

Как уже отмечалось, большинство учащихся сталкивались с трудностями при освоении действий по копированию группы строк. Первая причина этого заключается, по-видимому, в сложности алгоритма копирования (это подтверждается данными «экспресс-анкетирования»). Копирование группы строк в редакторе СТР осуществляется по следующему алгоритму:

- указать курсором первую строку копируемой группы;
- нажать CTRL b;
- указать курсором последнюю строку копируемой группы;
- нажать CTRL B;
- указать курсором место вставки;

вставить группу строк с помощью CTRL q.

Действительно, даже хорошо усвоив общий принцип действий по копированию текста, легко ошибиться при выборе соответствующих буквенных клавиш. Никакой смысловой связи выполняемых действий с латинскими буквами b, B, и q учащиеся не усматривают. По сути дела, они должны эти буквы просто выучить.

Вторая причина состоит в отсутствии простого способа указания места вставки группы строк. Если указать курсором определенную строку и нажать CTRL q, то возникает неопределенность: вставится ли текст выше указанной строки или ниже ее? Аналогичные проблемы возникают и при вставке отдельных символов.

Наблюдение за учащимися показали, что эффективность выполнения копирования снижается также по третьей причине, имеющей особую природу. Если первые две причины были связаны с отсутствием у учащихся некоторых необходимых знаний, то третья состоит в том, что в процессе обучения спонтанно появляются избыточные представления о системе. Было обнаружено, что многие учащиеся прежде чем фиксировать первую строку копируемого участка текста, переводят курсор в первую позицию этой строки. Таким образом, хотя учащиеся заранее предупреждались о том, что при указании строки курсор может находиться в любой ее позиции, они представляли себе, что для компьютера «имеет значение», какая часть строки находится справа от курсора. Объяснить это часто встречающееся заблуждение можно тем, что учащиеся переносили принципы естественной работы с текстом (и возможно, опыт работы с командами разрезания и склейки строк) на ситуацию, в которой действуют хотя и очень простые, но искусственные правила.

Предварительные требования к методике обучения работе с текстовым редактором

Опыт проведенного нами эксперимента позволяет сформулировать основные требования к учебному процессу, направ-

ленному на освоение учащимися навыков работы с текстовым редактором.

1. На начальном этапе должна быть сформирована познавательная мотивация, направленная на изучение текстового редактора. Для этого следует давать учащимся задания по подготовке текстов, которые сами по себе привлекательны для них (с этой целью необходимо правильно подобрать содержание текстов), но требуют повторения рутинных действий и преобразования готовых блоков.

2. На следующем этапе учащиеся должны не только хорошо усвоить значение команд редактора (для чего целесообразно опираться на опыт первого этапа — проблемного введения), но и наглядно представлять себе механизм их выполнения. С этой целью необходимо в наглядной форме представить учащемуся модель исполнителя, функционально эквивалентного изучаемому редактору.

3. На третьем этапе должна происходить отработка навыков. Это центральный и наиболее ответственный этап обучения. Учебный процесс должен быть здесь высокоиндивидуализированным. На этом этапе необходимо решение следующих задач:

диагностика ошибочных представлений учащегося о функциональной структуре текстового редактора как исполнителя с фиксированной системой команд;

коррекция выявленных ошибочных представлений;

достижение критического уровня скорости и точности использования команд текстового редактора.

4. Необходимым компонентом обучения навыкам работы с текстовым редактором является закрепление этих навыков на естественном материале. Это необходимо, чтобы учащиеся осознали возможности и границы применения текстовых редакторов для компьютерной подготовки текстов. Целесообразно использовать задачи, решение которых является важным для изучения других школьных предметов. Возможных вариантов здесь очень много. Например, учащимся может быть предложен текст, заведомо содержащий ошибки. Исправ-

ление этих ошибок (т. е. практическое применение знаний по русскому языку, если ошибки орфографические, либо литературе, географии, физике и т. д., если ошибки содержательные) позволит на конкретных примерах понять преимущества текстовых редакторов. Наиболее эффективным способом практического закрепления навыков является подготовка учащимися собственных текстов (сочинений, лабораторных работ и т. д.).

Практика показывает, что программное обеспечение учебного процесса по формированию навыков работы с текстовыми редакторами не должно сводиться только к самим этим редакторам (хотя проблема создания образца, удовлетворяющего требованиям, ждет своего решения). Необходимыми педагогическими программными средствами являются также специальные обучающие программы-тренажеры. Вопросы, связанные с разработкой перспективной модели текстового редактора и программ-тренажеров, будут обсуждены ниже.

Перспективы разработки программных средств

Обратимся к текстовым редакторам, которые были использованы нами в экспериментальном обучении. По сути дела, специально для целей школьного обучения предназначен лишь «лосън». Некоторые его недостатки отражают лишь неудачу при реализации совершенно верных и интересных идей. Так, низкие функциональные возможности — следствие опоры лишь на базовые, универсальные команды, характерные для всех типов текстовых редакторов. Неудобное двухполосное представление текста, имеющего ширину более 40 символов, определяется желанием сохранить крупный, хорошо различимый шрифт, а многострочная, трудная для понимания система помощи говорит о стремлении выйти за рамки обычной для профессиональных текстовых редакторов сверхлаконичной «шпаргалки». Идеи, положенные в основу редактора «лосън» (ограничение функциональных возможностей базовыми командами, 40-символьный экран для машин с малым раз-

решением, развитая система помощи), следует признать перспективными. Именно на них должна строиться, по нашему мнению, разработка учебного редактора текста. Практика показала, что ни адаптация специализированного текстового редактора, предназначенного для программистов, ни простое использование даже лучших образцов профессиональных текстовых редакторов широкого назначения не решают проблемы массового обучения в школе и ПТУ компьютерной подготовке текстов.

По нашему мнению, мысленная модель компьютерного текста и текстового редактора как исполнителя, выполняющего преобразование этого текста должна быть как можно более простой и естественной. Безусловно, нельзя опираться только на те представления, которые имеются у учащихся к началу работы с текстовыми редактором. В традиционном представлении текст зафиксирован на бумаге и потому неизменен. Учащимся необходимо объяснить и дать им возможность понять на практике, что текст, хранящийся в памяти ЭВМ, может легко преобразовываться по определенным правилам. Но традиционное представление состоит в том, что текст — это письменно зафиксированное определенное содержание и менять это представление нет никаких оснований. Другими словами, работа с текстовым редактором не должна строиться на понятии текста как совокупности строчек или других формальных характеристик. Формальные характеристики должны быть, насколько это возможно, скрыты от обучаемого, чтобы он мог максимально сосредоточиться на смысловых преобразованиях текста.

Так, смысловая перестановка в тексте — это не копирование группы строк с последующим уничтожением их в исходном участке текста, а перенос последовательности слов, несущих определенное содержание, в новое место с сохранением формата текста.

Именно исходя из требований естественности содержательного преобразования текста и должны быть определены базовые команды редактора. Возможен следующий набор команд:

выделение любого участка текста как последовательности символов между двумя выделенными символами (блока);
удаление блока (при этом оставшийся текст стягивается с сохранением заданного формата);

перенос блока в заданную точку (при этом текст растягивается, оставаясь в том же формате);

запоминание блока с его удалением (записи в буферное хранилище);

просмотр и редактирование содержания буферного хранилища;

перестановка двух блоков;

форматирование (изменение абзацного отступа, длины строки до 40 символов, числа строк на странице, нумерации страниц);

поиск по образцу;

отмена предыдущей команды.

Разумеется, наряду с этими командами сохраняются команды вставки, удаления и забоя символов, выполняемые с помощью специальных клавиш. Кроме того, целесообразно использовать пиктограммы для обозначения состояния редактора (режим вставки или замены, работы с текстом или содержанием буферного хранилища) и значений функциональных клавиш.

Примерное распределение команд по функциональным клавишам:

F1 — выделение границ блока;

F2/F7 — удаление/перенос блока;

F3/F8 — запоминание/воспроизведение блока;

F4/F9 — поиск по образцу/форматирование;

F10 — отмена предыдущей команды.

Целесообразно как можно шире использовать принцип умолчания. Так, одна и та же клавиша может использоваться для вставки блока в то место текста, где находится курсор, если имеется один блок, и для перестановки кусков текста, если имеются два блока. Для поиска по образцу также может быть использована одна клавиша: первое нажатие приводит к вводу образца, второе — к поиску до первой встречи образца в тексте, третье — к продолжению поиска и т. д. Единственная команда, выполнение которой требует перехода в командный режим, — форматирование. После первого нажатия на эту клавишу

происходит редактирование шаблона форматирования текста, а после второго нажатия — форматирование и переход в режим редактирования самого текста. Очень важной командой, обеспечивающей раскрепощенность обучаемого, является отмена предыдущего действия. Для текстового редактора, предназначенного для начинающих, нецелесообразно использовать такую сложную систему «откатки», которая использована в редакторе «Микромир», так как многие учащиеся не могут освоить эту команду, а у тех, кто ее освоил, снижается самоконтроль (эти учащиеся считают, что нет необходимости быть внимательным, если любые ошибки исправимы и любые результаты восстановимы).

Выше описаны требования к центральному звену обучения с текстовым редактором — самому редактору. Однако программное обеспечение данного раздела учебного процесса ни в коем случае не должно, по нашему мнению, ограничиваться только редактором. Необходима разработка еще как минимум двух компонентов программного обеспечения.

Первым является модуль, предназначенный для преподавателя. Этот модуль обеспечивает обработку (по заданному преподавателем алгоритму) текста, подготовленного учеником. Он должен настраиваться (программироваться) преподавателем и рассылаться вместе с текстовым редактором на рабочие места учеников. Функциональные возможности модуля еще предстоит уточнить, однако уже сейчас очевидно, что множество команд этого исполнителя должно включать:

поиск в тексте заданных образцов;
подсчет количества символов в тексте;
выдачу на экран сообщений ученику;
пересылку информации на рабочее место учителя по локальной сети.

Примером работы проверочного модуля может быть следующая воображаемая ситуация. Ученикам разослан текст, содержащий ряд орфографических ошибок, которые нужно исправить. Выполнив задачу, учащиеся включают проверочный модуль. Тот «ищет» ошибочно напечатанные слова.

Даже самая примитивная возможность работы проверочного модуля — посимвольное (с точностью до пробела) сличение текста, набранного учащимся, с текстом-образцом позволяет проводить «компьютерные диктанты».

Вторым необходимым компонентом системы программных средств, обеспечивающих эффективное обучение работе с текстовым редактором, является программа-тренажер. Эта программа должна выполнять такие функции, как диагностика уровня сформированности навыка работы с редактором, диагности-

ка ошибок и их коррекция с помощью системы подсказок и дополнительных заданий. В основу этой системы может быть положен анализ результатов освоения текстовых редакторов, который был проведен в нашем исследовании.

Описанный выше комплекс педагогических программных средств будет обладать, по нашему мнению, высоким потенциалом не только для формирования навыков диалога с компьютером, но и для развития письменной речи, а тем самым и общего развития познавательных процессов учащихся.

Г. АКимова

В. ДЕЗА,

канд. техн. наук

Демонстраторы учат программированию

Системный подход к проблемам конструирования курсов обучения приводит к необходимости определения концепции проектируемого курса. Если стоит вопрос о создании курса обучения началам программирования, то в настоящее время среди широко распространенных концепций явно выделяются три: преимущественное обучение некоторому языку программирования, обучение приемам алгоритмизации, повышение общего уровня знаний в области информатики и вычислительной техники с включением элементов программирования.

Авторы не склонны поддерживать какую-либо одну из них. Мы думаем, что использование неординарных методов может оживить любую. Вопрос заключается в том, насколько каждая «деформирует» процесс обучения. Много написано о гордиевом узле: обучать программированию достаточно эффективно можно учащихся, уже знакомых с основами алгоритмизации; для написания алгоритмов необходимо овладеть некоторым формальным языком; умение программировать позволяет упростить изучение новых формальных языков. Так замыкается круг. С чего же начинать?

В качестве собственной концепции авторы предлагают следующую: «Обучение методам конструктивного системного мышления и освоение навыков исполне-

ния операций, свойственных профессиональным программистам, независимо от языка программирования».

Такой подход накладывает отпечаток и на преподаваемый материал, и на средства компьютерной поддержки курса, и на методику проведения занятий: придется отказаться от предварительного обучения элементам алгоритмизации и от изучения формального языка программирования. Это становится возможным, если главенствующую роль возложить на компьютер, значительно расширить математическое обеспечение.

В каждом курсе обучения увязаны и функционируют совместно три субъекта: конструктор с задачей — продвижение в жизнь некоторой концепции обучения; преподаватель, реализующий творческие замыслы; учащийся, имеющий свою цель и готовый к восприятию знаний, но при наличии некоторых факторов: познание с минимумом затрат, познание «живого», интересного материала, познание в увлекательном процессе. На наш взгляд, в предлагаемом курсе найдены предпосылки «примирения» всех трех функционирующих субъектов.

Конструктор предлагает педагогу методику, согласно которой изучаются понятия целого как совокупности составляющих частей, декомпозиция целого на составляющие конструкции — блоки,

классификация блоков, организация межблочных связей, блок-схемы рассуждений, итерационные процедуры и т. п. Программирование же является той областью, которая поставляет задачи, необходимые для раскрытия содержания вводимых понятий; по мере усвоения приемов и методов системного мышления набор предлагаемых для решения задач изменяется.

Человек в повседневной жизни осуществляет формальную постановку различных задач, которые решает доступными средствами, алгоритмизирует их. Формулировка и формализация зависят в основном от окружающей обстановки и во многом ею диктуются. Помещение учащегося в фиксированную, специально сформированную среду обучения позволяет привязать его знания и умения к учебному процессу, активизировать познание. Такой подход можно назвать естественной схемой обучения.

Использование естественной схемы позволяет вводить многие понятия без точных определений. Например, говоря о перемещении «динамического изображения» по экрану дисплея, можно использовать такие понятия, как «движение», «структура поля перемещений», «шаг движения», «направление» и т. п. Все они известны учащимся; каждый имеет право понимать их по-своему. Для процесса познания надо сделать некоторое начальное усилие и представить себя частью среды — динамическим объектом, способным к перемещению (в соответствии с одним из принципов, предложенным С. Пейпертом).

Напомним, что в основу каждого Демонстратора положена «мягкая» конструкция, получающаяся при объединении некоторого количества исполнителей в единую систему. Демонстратор — совокупность исполнителей, связанных общим интерфейсом в комплекс, направленная на достижение конкретной методической цели (для этого используются идентификация, хранение, выбор и построение различных изображений на экране дисплея). В зависимости от контекста некоторые Демонстраторы можно рассматривать как банки данных, приспособленные для хранения изображений.

Демонстратор «Художник», достаточно подробно описанный в «ИНФО», № 3, 1990, позволяет в интерактивном режиме проводить алгоритмизацию задач методом проб и ошибок. Учащийся не замечает, что, ошибаясь и исправляя ошибки, он занимается алгоритмизацией задачи и построением алгоритма. Язык Демонстратора максимально приближен к русскому языку (с некоторыми вполне понятными сокращениями слов — операторов управления), поэтому учащийся не сразу осознает, что он уже пишет программу построения изображения. Таким образом достигается возможность обучения программированию без изучения основ алгоритмизации и языка. В процессе освоения различных способов построения изображений вырабатываются необходимые алгоритмизационные и языковые шаблоны.

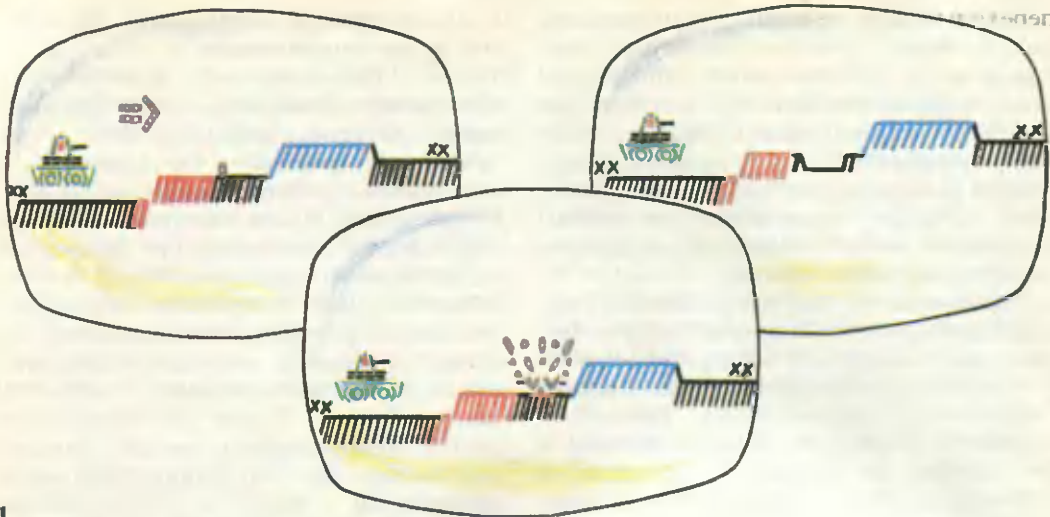
33

Демонстратор «Танк»

Изображения, высвечиваемые в процессе работы Демонстратора «Танк», подразделяются на три группы: а) статические изображения профилей пересеченной местности (дороги), б) динамическое изображение танка с поддержкой иллюзии его перемещения по пересеченной местности, в) синтетическое изображение сложного танкодрома, формируемое случайным образом из фрагментов различных статических изображений.

Задачи, решаемые с помощью Демонстратора «Танк», формулируются, как правило, в форме: «Провести танк до базы по танкодрому с номером n». При этом учащиеся ощущают себя водителями танка и решают задачи управления.

При работе с Демонстратором «Танк» от учащихся требуются умения, полученные при работе с Демонстратором «Художник»: проведение анализа задачи, разбиение ее на подзадачи (блоки), свободное владение системой отладки программ и т. п. Учащиеся обучаются составлению и отладке синтезированной главной подпрограммы (собирающей), умению отыскивать ошибки в программе с помощью специального пошагового



34 режима, организации ветвлений и другим умениям.

Усложнение задач происходит по двум направлениям: обучение преодолению стандартных препятствий и написание универсальных программ для прохождения с помощью одной программы нескольких дорог. Методически изучение прохождения стандартных препятствий связано с непрерывным усложнением программы преодоления. Для их построения требуются новые средства и приемы программирования, углубление знаний и привлечение новых механизмов анализа.

Знакомство с «Танком» начинается с задачи: «Проведение танка до базы по ровной дороге». Здесь осваиваются операторы управления пошаговым движением танка и способы организации простых циклов, обеспечивающих его перемещение. Затем на ровной дороге появляются препятствия в виде подъемов и спусков. Наличие разнородных препятствий позволяет ввести понятие блок-схемы и объяснить наглядно роль блок-схем в программировании.

Последняя из изучаемых дорог заминирована. Методикой предусмотрено, что алгоритм ее прохождения построит сами учащиеся. На рис. 1 изображены три последовательные стадии «ликвидации мины». Блок-схема создаваемой для «разминирования» программы достаточно разветвлена и требует применения аналитического и синтетического мето-

дов решения, а сама программа включает в себя сложные конструкции с вложенными реентерабельными блоками, условными и безусловными ветвлениями и т. п.

Язык управления Демонстратором «Танк» является расширением языка «Художника». Операторы движения здесь более разнообразны. На обложке изображена часть клавиатуры. Запомнить операторы движения достаточно просто, если мысленно поместить танк на цифру 5. Тогда станет понятным, почему оператор «6» обозначает движение на один шаг вправо, «4» — влево, «1» — влево вниз, «3» — вправо вниз, «7» — влево вверх, «9» — вправо вверх. Оператор «5» отведен для «выстрела».

Продвинуть танк по дороге до базы невозможно без информации о встречающихся на ней препятствиях. Ее поставляют два радара — передний и хвостовой. Непрерывно поступающая через них информация содержит два символа (из символов, составляющих дорогу): первый символ, расположенный непосредственно перед танком, и второй — за танком.

Можно коротко сказать, что переход к этому Демонстратору увязан с переходом от подпрограмм (основных программных единиц, вводимых с помощью «Художника») к условным предложениям, ветвлениям программ, итерационным циклам. В Демонстраторе, естественно, используются и безусловные

переходы: вводятся метки и обращения на них (передачи управления). Введение итерационных циклов, расширение возможностей блочного программирования за счет «вложенных блоков» и «реентерабельных блоков», организация условных предложений — все это позволяет учащимся перейти к новым формам программирования — блок-схемам как аппарату конструктивного мышления. Учащиеся получают возможность размышлять в терминах: «дойти до препятствия», «преодолеть подъем», «нейтрализовать мину» и т. п. При этом сама программная реализация нужных блоков представляется учащимся как нечто естественное, простое и легкодоступное.

Безусловно, составить и отладить программу управления движением танка до базы и обратно по случайному танкродому — задача довольно сложная. Поэтому значительная роль отводится систематизации процесса выработки алгоритма и конструкции создаваемой программы.

Демонстратор «Лабиринт»

Среда обучения — Лабиринт — широко распространена в курсах информатики и программирования. Лабиринт может служить полигоном для проверки способности учащегося к самостоятельному построению алгоритмов решений некоторых задач. Это последний Демонстратор, включенный в курс.

В Демонстраторе «Лабиринт» поддерживается среда, необходимая для изучения сложных логических выражений, ветвлений в зависимости от значения булевой переменной, строятся условные циклы.

После прохождения простых лабиринтов, в которых легко определить самый короткий путь, учащимся предлагается пройти по внешней и внутренней стороне границы квадрата. При прохождении такого псевдолабиринта учащиеся строят блоки, необходимые для движения вдоль внешних и внутренних сторон квадрата (восемь блоков). Их текст сохраняется на диске и может после этого использоваться в других программах. Этим блокам достаточно, чтобы «ежик» — динамический объект — нашел выход из лабиринта; необходимо,

правда, построить далеко не простую собирающую программу.

Видя изображение лабиринта, учащийся невольно использует эту информацию для построения программы. Это затрудняет построение универсального алгоритма прохождения классического лабиринта. Чтобы нейтрализовать влияние дополнительной информации, в Демонстраторе предусмотрено изменение внешнего вида лабиринта случайным образом при каждом запуске программы. Учащийся не знает, по какому лабиринту пойдет его «ежик», поэтому учащиеся должны составить универсальную программу.

Демонстратор «Лабиринт» имеет два равноправных языка управления: один из них (общий для всех Демонстраторов курса) именуется основным, а второй приближен к языку Бейсик: движения вверх (up), вниз (down), вправо (right), влево (left) считаются встроенными функциями. Это дает возможность использовать в обучении некоторое подмножество, точнее подязык, известного формального языка программирования. Демонстратор «Лабиринт» включает исполнители для поддержания трансляции с языка Бейсик на основной язык. Их введение позволяет упростить методику раскрытия материала о различных формальных языках программирования, об операции трансляции текстов, о структуре трансляторов и их работе, дает реальную возможность «увидеть» текст на Бейсике и результат трансляции его на основной язык.

Переход на Бейсик осуществляется практически за один урок построением таблицы соответствия операторов основного языка и Бейсика. В дальнейшей работе учащиеся сохраняют за собой право выбора того языка, на котором они предпочитают программировать. Интересен, на наш взгляд, тот факт, что учащиеся, как правило, выбирают таким языком Бейсик, считая его «настоящим языком программирования», хотя некоторые части задачи продолжают отлаживать на основном языке Демонстратора. Курс заканчивается после того, как учащиеся научатся писать универсальные

программы прохождения классического лабиринта на Бейсике.

Итак, описанный курс обучения началам программирования младших школьников представлен тремя различными геометрическими средами обучения, поддерживаемыми Демонстраторами «Художник», «Танк» и «Лабиринт». Каждая следующая среда методически продолжает предыдущую, а весь курс благодаря

системе Демонстраторов полностью построен на игровой мотивации.

Курс готов к тиражированию, и заинтересованные организации могут ознакомиться с порядком его приобретения.

Адрес для справок: 117312, Москва, проспект 60-летия Октября, дом 9, ВНИИСИ АН СССР, комната 618, Акимовой Г. П., Дежа В. Н.

В. УРНОВ, А. ЗУБЧЕНКО

Москва

Переходные среды в предмете «Информатика»

Проникновение компьютера в сферу образования остро выявило серьезные недостатки, свойственные традиционному обучению: неумение адаптироваться к быстроменяющемуся миру, отсутствие гибкости и ориентации на самостоятельность ребенка, формализм, воспитание у детей устойчивых фобий к отдельным школьным предметам и к процессу учения вообще. Неудивительно, что в этих условиях практика использования компьютера складывается как практика «программирования» учащегося, т. е. компьютер используется преимущественно в качестве инструмента накопления фрагментарного знания и натаскивания на частные умения и навыки.

К сожалению, подобная концептуальная ориентация превалирует в большинстве школьных курсов по информатике, ориентированных главным образом на узкотематическую направленность содержания, рекомендуемого к усвоению. Последнее практически полностью исключает возможность культурного переноса учащимися полученных знаний и умений в другие сферы человеческой деятельности.

Между тем школа должна давать человеку не просто свод некоторых сведений из области науки, а то, что называется философией науки или ее системной проекцией на школьную дисциплину. Каждый предмет — это новая

грань действительности, новый взгляд на мир, иной способ мыслить, ранее неизвестные возможности решений, наконец, новые или усовершенствованные интеллектуальные средства. Именно в этом заключается истинная ценность любого школьного курса.

Если говорить об информатике, то ее рейтинг как дисциплины, прививающей алгоритмический стиль мышления, достаточно высок. И все же рискнем утверждать, что такой стиль мышления не является базисным в современной цивилизации, в отличие, допустим, от содержательно-логического мышления или умения свободно проникать в семантику литературных ассоциаций или символизм изобразительно-пластических образов. Следовательно, информатика, о чем свидетельствует и современная мировая практика, не может замыкаться в узких рамках собственной специфической предметности. Она должна искать связь с другими предметами, проникать в более широкий спектр содержания; в противном случае ей грозит опасность превратиться в некоторую «закрытую» форму знания.

Настоятельные требования времени таковы, что школьным курсам необходимы учебники, пособия, методики, которые могли бы обеспечить как учителям (в первую очередь учителям информатики), так и школьникам возможность постепенного перехода к про-

цессу обучения в насыщенной компьютерной среде. Понятно, что новые технологии обучения проще внедрять в преподавание именно информатики, однако при этом надо обязательно думать о целенаправленном распространении этих технологий на другие предметы. С этой точки зрения информатика как школьный курс должна получить новое толкование, иную концепцию и соответственно иную цель.

Мы полагаем, что такие новые технологии обучения, которые должны составлять содержание предмета, называемого сегодня информатикой, следует определять как переходные среды.

Главной отличительной особенностью переходных сред является их ориентация на формирование не столько знаний, сколько средств, при помощи которых учащиеся могут самостоятельно добывать новые знания, систематизировать уже полученные и, кроме того, конструировать новые интеллектуальные орудия.

Примером такой переходной среды может служить система исполнителей, образующих пакеты обучающих программ. Однако, если эта система носит замкнутый характер, т. е. не позволяет легко расширять ее возможности, исправлять или видоизменять алгоритмы деятельности исполнителей, то ее ценность существенно понижается. Она поможет учащемуся несколько лучше и быстрее овладеть тем или иным понятием, чем при посредничестве преподавателя, но едва ли она обеспечит качественные условия для возникновения обобщенных интеллектуальных средств и их адекватного переноса.

Это означает, что определяющей конструктивно-моделирующей характеристикой переходной среды должна быть ее открытость. Иными словами, она должна представлять собой такую языковую среду, которая поддерживала бы некоторый набор исполнителей и обеспечивала бы их создание и модификацию средствами, вполне доступными учащимся. Основой может служить язык программирования — достаточно мощный и поддерживающий современные технологии программирования (Бейсик, разумеется, этим требо-

ваниям не отвечает).

Примером переходной среды может служить программная поддержка (авторы А. Г. Кушниренко и др.) «КуМир» — система исполнителей, основанная на интерпретаторе школьного алгоритмического языка. Состав исполнителей в этой среде может быть расширен, в том числе и самими учащимися, причем область применения исполнителей не ограничивается одной информатикой. Любой из исполнителей может быть без труда целенаправленно изменен для демонстрации свойств моделируемой предметной области. Педагогический эффект, который достигается благодаря возможностям таких переконструирований, чрезвычайно велик и едва ли достижим при помощи традиционных методических приемов. Если принять во внимание, что учащиеся X — XI классов в своей массе владеют школьным алгоритмическим языком, то при наличии соответствующих исполнителей эту среду можно использовать в различных предметах школьного курса, т. е. на очень широком спектре предметного содержания.

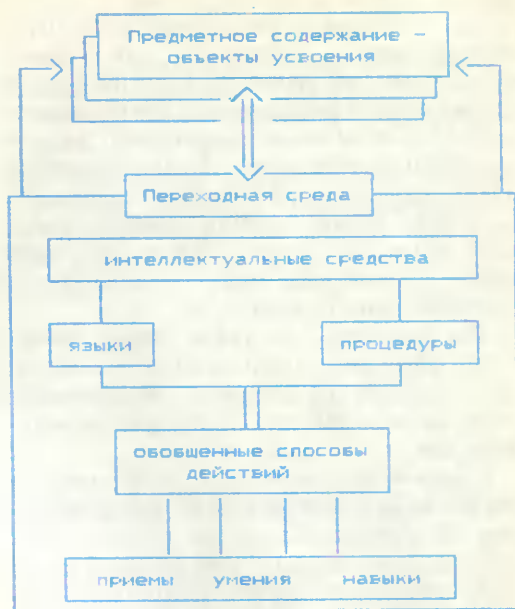
Другим примером новых технологий открытого типа может служить среда обучения ЛОГО, способная революционизировать методику преподавания математики, физики и многих других предметов. Отличительной особенностью среды ЛОГО (С. Пейперт) является установка на то, что ребенок должен сам «программировать и обучать» компьютер в процессе естественного и незатрудненного диалога с машиной, а по мере овладения сложнейшим полифункциональным устройством переходить к непосредственному использованию научных идей, математических и лингвистических понятий, прибегать к искусству интеллектуального моделирования.

В настоящее время философия ЛОГО — одно из немногих действительных интеллектуальных достижений в области методики обучения, связанной с новыми технологиями, — после четверти века экспериментирования начинает все шире входить в практику. Наличие в языке уникального средства ввода/вывода «Черепашка», наряду с

особенностями самого ЛОГО (во многом родственного с языком искусственного интеллекта ЛИСП), собственно и создает переходную среду, в условиях которой ребенок осваивает математические и физические идеи. Даже такие, считающиеся труднодоступными фундаментальные математические понятия, как дифференциал или кривизна, удастся психологически адекватно объективировать («показать» их сущностные или внутренние характеристики) с помощью Черепашки. Что же касается более простых понятий и представлений, то они могут быть открыты детьми, погруженными в среду ЛОГО, совершенно самостоятельно. Немаловажно также и то, что возможность легко подчинить Черепашку своей воле и задать этому динамичному объекту определенное поведение позволяет ребенку рефлексировать свои собственные шаги и, наоборот, осознание своего поведения выступает для него как источник содержательных «теорий» и идей для общения с Черепашкой и как основание для выбора адекватных управляющих воздействий — планирования своих действий. Иными словами, благодаря работе в среде ЛОГО дети обретают уверенность в своих силах и могут выразить свое знание предмета в форме, отличной от вербальной (превалирующей при обучении естественным наукам в сегодняшней школьной практике), т. е. с помощью собственных намеренно планируемых и моделируемых действий.

Существо новых технологий как переходных сред, направленных прежде всего на создание условий, благодаря которым учащиеся могут успешно осваивать, конструировать и модифицировать средства учения и самообучения, можно представить в виде некоторой системы составляющих, способной выразить, по нашему мнению, наиболее адекватную на сегодняшний день философию построения школьного курса по информатике (см. рис.).

При таком подходе полностью или почти полностью меняются методическая направленность и соответственно само содержание учебника, концепция поддерживающего его программного продукта.



Эта ориентация означает, что учебник информатики должен обеспечивать условия, которые позволяют учителю при помощи определенного языка (языков) переходных сред транслировать широкий спектр интеллектуальных средств. Для учащегося это означает освоение этих средств на различном предметном материале посредством решения доступного разнообразия предметно-практических задач. Итак, в качестве первого реального шага дети должны овладевать не узконаправленными процедурами кодирования и формализации (алгоритмизации), а способами и приемами культурного мышления, дающими возможность преобразовывать фрагментарные сведения и непосредственные знания в осмысленную информацию. Последняя, в свою очередь, включаясь в анализ, описание и моделирование предметного содержания, сама становится новым интеллектуальным средством в арсенале уже имеющихся у ученика.

Для реализации этого подхода учебник информатики должен быть поддержан профессионально исполненным программным обеспечением, которое могло бы надежно эксплуатироваться на большей части имеющихся в школе компьютеров.

Вы учите человека общаться с компьютером? Мы учим компьютер общаться с человеком!

Фирма «МИГ»

предлагает обучающие программы для КУВТ-86, ДВК, БК-0010/11:

MICROB — Бейсик для начинающих (60/150 руб. — в числителе стоимость программ для школ и ПТУ, в знаменателе — для остальных покупателей);

СИ — полный языковый пакет для ДВК и БК-0011 (60/130 руб.);

PAINT — учебный графический редактор с подсказками (60/150 руб.);

PDP11 — действующая модель процессора ЭВМ ряда СМ, ДВК, БК — отличный тренажер для осваивающих ассемблер PDP11 (90/600 руб.);

БЭБИ-САПР — учебная графическая САПР РЭА, позволяющая создавать несложные (до 16 RS-, JK-, D-триггеров) цифровые схемы (90/600 руб.).

Если вы желаете учить ваших подопечных всему сразу, вас может устроить только обучающая система «Незнайка», отмеченная дипломом II степени на Всесоюзном конкурсе программ для ПК!

(КУВТ-86/Бейсик & Фокал; 420/2500 руб.)

Фирма «МИГ» примет заказы на программные комплексы произвольного назначения для КУВТ-86/86М по государственным расценкам.

Вашего звонка ждут по телефонам:

(095) 375-15-09; (095) 327-94-59 (Москва).

Учебно-методический центр

Московского городского института усовершенствования учителей
приглашает

на курсы по методике преподавания информатики по машинному варианту.

Занятия проводятся преимущественно в каникулярное время

в течение двух недель:

ноябрь 1990 г., январь 1991 г., март 1991 г., июнь 1991 г. и т. д.

Обучение включает:

теоретические и практические занятия по научно-методическим основам преподавания курса информатики по машинному варианту;

обучение работе на различных видах ВТ: (КУВТ - 86 с Бейском и Фокалом, УКНЦ, «Корвет», «Агат», «Ямаха», IBM совместимые компьютеры), знакомство с программным обеспечением и особенностями методики преподавания курса ОИВТ на данном виде ВТ.

Слушателям курсов передается программно-методический комплекс по преподаванию информатики по учебнику под редакцией В. А. Каймина, включающий в себя учебник, методическое пособие для учителей (1 и 2 части), задачник, программное обеспечение в поддержку курса.

Слушатели курсов обеспечиваются общежитием.

Желающие прослушать курс должны прислать гарантийное письмо об оплате с указанием желательного времени обучения. Учебно-методический центр вышлет приглашение и счет. Оплата производится до начала занятий по безналичному расчету.

Адрес МГИУУ: 125167, Москва, Авиационный пер., д. 6, кабинет информатики.

Проезд: м. «Аэропорт».

Телефон для справок: 151-59-31.

Лартор — новая ступень

Так называемые портативные (лартор) компьютеры, раньше имевшие размеры кейса и вес более 5 кг, наконец перешли рубикон. Новые модели фирмы Compaq (США) LTE и LTE/286 весят по 2,5 кг и имеют размеры 27×22×5 см. В этих «книжного» размера корпусах поместились 3,5-дюймовые дисководы и электролюминесцентные экраны CGA. В основе LTE, стоящего 2399 долларов, микропроцессор 80С86; в LTE/286 (3899 долларов) — 80С286.

Еще одну модель этого класса выпустила фирма Radio Shack (США). Это Tandy 1100 FD. Его вес 3 кг, размеры 30×25×6 см, зато цена всего 999 долларов. Есть дисковод для 3,5-дюймовых дисков, жидкокристаллический черно-белый экран. Интересен большой набор записанных в ПЗУ программ: от Desk-Mate Graphical User Interface до текстового процессора со словарем на 90 000 слов (для проверки правописания).

У нас есть шанс!

По мере решения проблем с вычислительной техникой для школ все острее встает проблема программного обеспечения. К сожалению, в отчетах гораздо эффективнее выглядят данные о числе кабинетов ВТ, чем о количестве закупленных для них программ; а уж качество и вовсе «не отчетная» материя... И так обстоят дела не только у нас. В 1987/88 учебном году американские школы потратили миллиард долларов на компьютеры и только 210 млн. долларов на программы, в то время как обычное соотношение стоимости «железа» и «начинки» — 1:1. А что если взяться за ум и уделить должное внимание программному обеспечению — глядишь, придется американцам нас догонять!

Что купить?

Если вы хотите купить принтер, а какой выбрать — не знаете (бывает и такое с обладателями свободно конвертируемой валюты), посмотрите, как поступают другие. Наиболее популярными матричными принтерами в 1988 г. были Apple Image Writer II (про-

ЧТО МОЖЕТ ЭВМ

дано более 97 тыс. штук), Epson LX 800 (почти 49 тыс.), Panasonic 1080 i (столько же), Star Micronix NX 1000 и Panasonic 1090 i (по 41 700). Суммарный объем продаж, как видите, невелик. Видимо, матричные принтеры выходят из моды. Действительно, лазерные принтеры покупают гораздо активнее: HP Laser Jet II — 430 тыс., Apple Laser Writer II NT — более 100 тыс.

26М байт в нагрудном кармашке

Любой программист знает, что 26М байт информации сегодня — это в лучшем случае два десятка дискет или увесистая металлическая коробка винчестера.

Фирма «Ситизен уотч» в скором времени заставит изменить ставшие уже привычными представления. Разработанный ее сотрудниками дисковод для гибких магнитных дисков диаметром 3,5 дюйма (89 мм) позволяет записывать до 26,5М байт информации на один диск. Сам дисковод тоже довольно компактен. Его размеры всего 101,6×157×25,4 мм при весе 470 г, а точная механика обеспечивает запись 540 дорожек на дюйм при скорости 600 оборотов в мин.

Дисковод имеет неплохие динамические характеристики. В среднем на одно обращение к диску дисковод тратит всего около 50 миллисекунд, а скорость его обмена информацией с компьютером достигает 3,58М бит в сек

Изображение без помех

Компьютеры и видеотехника настолько взаимопроникли друг в

друга, что сейчас уже трудно сказать, где какие технологические новшества зародились впервые. Высококачественные мониторы, магнитная лента, видеодиски, микропроцессоры...

Японская фирма «Панасоник» начала производить компактное видеоустройство, больше напоминающее переносной компьютер, чем видеосистему. А виной тому плоский цветной жидкокристаллический монитор, в закрытом положении служащий крышкой компактного видеочемоданчика. Впервые такие дисплеи промышленно начали использовать в переносных компьютерах. Но появление так называемых активных матричных ЖК-дисплеев, каждый пиксел поля изображения которых снабжен активным элементом, позволил применить их в видеотехнике.

Самый молодой миллиардер в мире

Уильям Гейтс, основатель фирмы по производству программного обеспечения Microsoft. Ему 33 года, в списке богатейших людей 1989 г. он занимает 131-е место. Вот что может ЭВМ!

Своим успехом Гейтс обязан в равной, пожалуй, мере везению и таланту. Научившись программировать в 13 лет (талант? Безусловно. Повезло? А как же! Двадцать лет назад и в США ЭВМ были не очень-то доступны), он еще учился в школе, когда в продаже появился первый персональный компьютер «Альтаир 8800» — машина с ОЗУ 126 байт, без клавиатуры и дисплея (информация вводилась побитно путем манипулирования переключателями, выводилась аналогично с помощью двух десятков лампочек), без внешней памяти (впрочем, перфокарочный ввод у «Альтаира» был). Герою нашей истории вместе с его другом П. Алленом пришлось в голову написать для «Альтаира» интерпретатор Бейсика. Эта программа имела полный успех, была первой коммерческой программой для персонального компьютера и положила начало коммерческо-программистской деятельности, принесшей 15 лет спустя такие внушительные плоды.

А фирма Microsoft тем временем заняла в 1988 г. первое место среди производителей программного обеспечения, получив прибыль в 719 млн. долларов.

С. ГРИГОРЬЕВ

Йошкар-Ола

Работа системы Пролог-Д

Определяющим направлением развития информатики является создание технологии обработки знаний. Особую роль в этом процессе призваны сыграть языки логического программирования, в частности Пролог. Использование Пролога в народном образовании СССР связывается и с новым учебником информатики, и с широкими перспективами в изучении логики, элементов искусственного интеллекта, становящихся все более актуальными.

Ранее считалось, что транслятор с Пролога реализуем только на мощных компьютерах, имеющих высокое быстродействие и большие объемы памяти. Ограниченность возможностей школьных машин была серьезным препятствием на пути внедрения систем логического программирования в народное образование. Однако путем некоторого изменения синтаксиса входного языка и совершенствования алгоритмов удалось разработать компактную и эффективную версию системы логического программирования Пролог-Д для самых массовых учебных ЭВМ, используемых в СССР. В настоящее время полностью завершена разработка Пролога-Д для КУВТ «Ямаха MSX-2», КУВТ-86, «Электроника БК-0010 (.01)» с различными устройствами внешней памяти (магнитофон, кассета ПЗУ). В стадии завершения — работы на ПЭВМ «Электроника УКНЦ», «Электроника БК-0011», «Ямаха MSX-1».

Данная статья открывает цикл, посвященный принципам работы с системой Пролог-Д. Она посвящена начальному этапу — работе с редактором и запуску системы. Описываются две версии Пролога-Д: для MSX-2 и БК-0010. Версии отличаются друг от друга некоторыми возможностями и деталями интерфейса, что обусловлено разными клавиатурами и техническими возможностями ПЭВМ, поэтому статья (как и весь цикл)

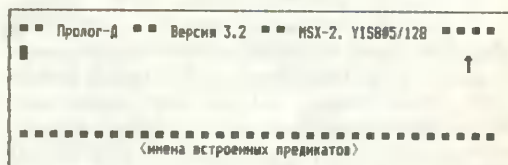
будет разделена на две части: одна описывает версию 3.2 для ПЭВМ «Ямаха MSX-2», а вторая — для КУВТ-86, «Электроника БК-0010 (.01)».

Система логического программирования Пролог-Д располагает достаточно развитыми средствами услуг пользователю. Целесообразно перед началом знакомства с языком освоить принципы работы с системой.

I. Пролог-Д MSX-2, версия 3.2

Информация на экране дисплея

После загрузки системы Пролог-Д на экране появляется заставка, а затем — окно исходного текста Пролог-программы.



Окно исходного текста используется для записи и редактирования программ на Прологе-Д. Для редактирования используется встроенный экраный редактор, его директивы приведены ниже. Отмеченные на рисунке стрелкой четыре квадратика в верхней строке представляют собой указатель, показывающий заполнение базы знаний. Положение указателя полностью заполненной базы знаний на рисунке отмечено точкой.

Редактор системы

Текст программы можно ввести с клавиатуры либо с внешнего устройства (дисковода).

Для ввода текста с клавиатуры можно пользоваться всеми алфавитно-цифровыми

клавишами. В тексте может быть любое количество пробелов, они игнорируются. Однако длина строки не более 128 символов. Ввод предложения осуществляется нажатием клавиши RETURN. Обязательное условие: в одной строке должно быть только одно предложение Пролога-Д.

Для редактирования текста программы используются управляющие клавиши стандартной клавиатуры ПЭВМ. Список выполняемых ими функций приведен в таблице.

| Действие | Кодирование |
|---|-------------|
| Перемещение по тексту | 〈стрелки〉 |
| Перемещение в начало текста | Ctrl+K |
| Перемещение в конец текста | Ctrl+V |
| Перемещение в конец предложения | Ctrl+N |
| Уничтожить символ слева от курсора | BS |
| Уничтожить символ в позиции курсора | DEL |
| Уничтожить текст базы знаний | Ctrl+C |
| Уничтожить предложение | Ctrl+U |
| Уничтожить символы до конца предложения | Ctrl+E |
| Включение/выключение режима вставки | INS |
| Вывод текста программы на экран | Ctrl+T |
| Печать текста базы знаний с остановкой | Ctrl+T+stop |

В данной версии системы Пролог-Д не предусмотрена «сборка мусора» при редактировании файлов, в результате может возникнуть переполнение памяти при длительном редактировании. Поэтому целесообразно пользоваться традиционными редакторами текстов с последующим вводом программ с дисководов.

Система Пролог-Д предполагает использование встроенных предикатов. Их имена запрограммированы на специальных клавишах, обозначаемых символами F2—F10. Такое использование клавиш существенно облегчает работу с системой. Особое значение имеет клавиша F1. Ею кодируется имя последнего введенного предложения программы.

Работа с дисководом допускает три режима: чтение файла, запись файла, просмотр файлов на диске.

Для чтения файла с диска нужно нажать клавиши Ctrl+L и в ответ на приглашение **Чтение.** Имя файла: {〈имя дисковода〉:} 〈Имя файла〉 ввести имя нужного файла.

Обратите внимание: при чтении файла старое содержимое базы знаний пропадает. Для создания файла можно пользоваться не только редактором системы Пролог-Д, но и другими текстовыми редакторами.

Для записи файла на диск нужно нажать клавиши Ctrl+S и в ответ на приглашение **Запись.** Имя файла: {〈имя дисковода〉:} 〈Имя файла〉 ввести имя нужного файла. База знаний записывается на диск целиком.

Для просмотра директория достаточно нажать клавиши Ctrl+D и в ответ на приглашение

Просмотр файлов: {〈имя дисковода〉:} 〈Имя файла〉 ввести имя нужного файла (файлов).

Для завершения работы с системой Пролог-Д нужно нажать клавиши Ctrl+Q (будьте внимательны — результаты работы безвозвратно теряются!).

Примеры работы с редактором

Исправление неверного символа.

```

■ ■ Пролог-Д ■ ■ Версия 3.2 ■ ■ MSX-2, Y1S805/128 ■ ■ ■ ■
бабушка(x,y)←-мама(x,z),мама(z,y);
бабушка(x,y)←-мама(x,z),pапа(z,y);

```

В подчеркнутом слове необходимо заменить латинскую букву *p* на русскую букву *п*. Для этого надо установить курсор под этим символом, нажать INS, набрать правильные символы, нажать RETURN, вновь нажать INS.

Удаление неверного символа.

```

■ ■ Пролог-Д ■ ■ Версия 3.2 ■ ■ MSX-2, Y1S805/128 ■ ■ ■ ■
бабушка(x,y)←-мама(x,z),мама(z,y);
бабушка(x,y)←-мама(x,z),pапа(z,y);

```

В подчеркнутом слове необходимо удалить латинскую букву *p*. Для этого необходимо установить курсор под этим символом, нажать DEL и RETURN.

Вставка символа.

```

■ ■ Пролог-Д ■ ■ Версия 3.2 ■ ■ MSX-2, Y1S805/128 ■ ■ ■ ■
бабушка(x,y)←-мама(x,z),мама(z,y);
бабушка(x,y)←-мама(x,z),апа(z,y);

```

В подчеркнутом слове необходимо вставить букву *п*. Для этого необходимо установить курсор под этим символом и нажать **п** и RETURN.

Удаление предложения.

```

■ ■ Пролог-Д ■ ■ Версия 3.2 ■ ■ MSX-2, Y1S805/128 ■ ■ ■ ■
бабушка(x,y)←-мама(x,z),мама(z,y);
бабушка(x,y)←-мама(x,z),папа(z,y);

```

Подчеркнутое предложение необходимо удалить. Для этого необходимо установить курсор на удаляемую строку и нажать **Ctrl+U**. В результате получится:

```
■ ■ Пролог-Д ■ ■ Версия 3.2 ■ ■ MSX-2, VIS005/128 ■ ■ ■ ■
бабушка(x,y)←-мама(x,z),мама(z,y);
```

Удаление текста до конца предложения.

```
■ ■ Пролог-Д ■ ■ Версия 3.2 ■ ■ MSX-2, VIS005/128 ■ ■ ■ ■
бабушка(x,y)←-мама(x,z),мама(z,y);
бабушка(x,y)←-мама(x,z),папа(z,y);комментарий
```

В подчеркнутом предложении необходимо удалить слово «комментарий». Для этого необходимо установить курсор перед удаляемым словом и нажать **Ctrl+E**.

Запуск системы

Чтобы проверить работоспособность системы Пролог-Д, достаточно набрать вопрос — предложение, начинающееся с символа «?». Проще всего составить его из встроенных предикатов. Они предназначены для выполнения арифметических действий, вывода графики и других подобных операций. Встроенные предикаты, если они записаны в вопросе, должны выполняться одинаково независимо от того, записана в память машины база знаний или нет и какая это база знаний. В определенном смысле это напоминает непосредственный режим работы в Бейсике. Например, если необходимо построить на экране отрезок, соединяющий точки с координатами (10, 10) и (100, 100), то достаточно задать вопрос:

?линия(10, 10, 100, 100, 1);

Требуемый отрезок появится на экране. Для возврата в редактор достаточно дважды нажать на любую клавишу.

Если необходимо вычислить число $x=2 \times 3+1$, то для этого достаточно набрать вопрос:

?УМНОЖЕНИЕ(2, 3, 1, x);

Ответ системы:

x=7
ДРУГИХ РЕШЕНИЙ НЕТ

Чтобы попробовать более сложные примеры работы системы Пролог-Д, рекомендуется набрать текст такой базы знаний:

```
мама(Люся, Саша);
папа(Сергея, Саша);
бабушка(Надя, Саша);
внук(x, y)←-бабушка(y, x);
```

Ей можно задать вопрос

?мама(x, Саша)

означающий «Как зовут маму Саши?». Ответ системы:

x=Люся
ДРУГИХ РЕШЕНИЙ НЕТ

Нажмите на любую клавишу, и база знаний вновь появится на экране. Можно попробовать задать еще несколько вопросов, например:

```
?бабушка(x, y);
?vнук(x, y)
?папа(Сергея, x);
```

О том, почему получаются именно такие ответы, будет рассказано в дальнейшем.

II. Пролог-Д «Электроника БК-0010» (КУВТ-86)

43

Информация на экране дисплея

После загрузки системы на экране появляется заставка, а затем — окно исходного текста Пролог-программы.

| | |
|--|-----|
| • Пролог-Д 1.0 • Редактирование • КТ+? Подсказка | Лат |
| | |

Окно исходного текста используется для записи и редактирования программ на Прологе-Д. Для редактирования используется встроенный экраный редактор.

Для индикации результатов работы используется окно вывода результатов.

| | |
|-----------------------------|-----|
| • Пролог-Д 1.0 • Выполнение | Лат |
| | |

Окно результатов выводится на экран автоматически после ввода вопроса.

Для возврата в окно исходного текста достаточно нажать любую клавишу.

Редактор системы

Работа с клавиатурой отличается от Пролога-Д для MSX-2 максимальной длиной строки (64 символа) и наличием подсказки — нажатие клавиш КТ и ? вызывает на экран список функций управляющих клавиш.

Для чтения файла с ленты нужно нажать

клавиши **КТ+Ч** и в ответ на приглашение **Имя?**

вести имя нужного файла. При чтении файла прежнее содержимое базы знаний уничтожается. Для подготовки файла можно пользоваться не только редактором системы Пролог-Д, но и другими текстовыми редакторами.

Для записи файла на ленту нужно нажать клавиши **КТ+З** и в ответ на приглашение **Имя?**

вести имя нужного файла. База знаний записывается на ленту целиком.

Для завершения работы с системой Пролог-Д нужно нажать клавиши **КТ+К**.

Примеры работы с редактором

Исправление неверного символа.

44

| Пролог-Д 1.0 | Редактирование | КТ+? | Подсказка | Лат |
|---|----------------|------|-----------|-----|
| <pre>бабушка(x,y)←-мама(x,z),мама(z,y); бабушка(x,y)←-мама(x,z),<u>рапа</u>(z,y);</pre> | | | | |

В подчеркнутом слове необходимо заменить латинскую букву *p* на русскую букву *п*. Для этого необходимо установить курсор справа от буквы *p*, нажать клавиши **←←**, **→→**, **РУС**, набрать правильный символ *п*.

Удаление неверного символа.

| Пролог-Д 1.0 | Редактирование | КТ+? | Подсказка | Лат |
|--|----------------|------|-----------|-----|
| <pre>бабушка(x,y)←-мама(x,z),мама(z,y); бабушка(x,y)←-мама(x,z),<u>апа</u>(z,y);</pre> | | | | |

В подчеркнутом слове необходимо удалить латинскую букву *p*. Для этого необходимо установить курсор справа от буквы *p* и нажать клавишу **→**.

Вставка символа.

| Пролог-Д 1.0 | Редактирование | КТ+? | Подсказка | Лат |
|--|----------------|------|-----------|-----|
| <pre>бабушка(x,y)←-мама(x,z),мама(z,y); бабушка(x,y)←-мама(x,z),<u>апа</u>(z,y);</pre> | | | | |

В подчеркнутом слове необходимо вставить букву *п*. Для этого необходимо установить курсор под этим символом, нажать клавишу **→** и нужную символную.

Уничтожение текста программы.

Нажать одновременно клавиши **СБР** и **РП**. Удаление предложения.

| Пролог-Д 1.0 | Редактирование | КТ+? | Подсказка | Лат |
|--|----------------|------|-----------|-----|
| <pre>бабушка(x,y)←-мама(x,z),мама(z,y); бабушка(x,y)←-мама(x,z),<u>апа</u>(z,y);</pre> | | | | |

Подчеркнутое предложение необходимо удалить. Для этого необходимо установить курсор в начало удаляемой строки и нажать клавиши **СБР** и **→**.

Удаление текста до конца предложения.

| Пролог-Д 1.0 | Редактирование | КТ+? | Подсказка | Лат |
|---|----------------|------|-----------|-----|
| <pre>бабушка(x,y)←-мама(x,z),мама(z,y); бабушка(x,y)←-мама(x,z),<u>апа</u>(z,y);комментарий</pre> | | | | |

В подчеркнутом предложении необходимо удалить слово «комментарий». Для этого необходимо установить курсор перед удаляемым словом и нажать клавиши **СБР** и **→**.

Запуск системы

Проверка работоспособности системы производится так же, как и на **MSX-2**. Отличия — в синтаксисе. Например, если необходимо построить на экране отрезок, соединяющий точки с координатами (10, 10) и (200, 200), то нужно задать вопрос:

?ОТР(10,10,100,100,1);

Если необходимо вычислить число $x = 2 \times 3 + 1$, то нужно задать вопрос:

?ВЫЧ(2,3,1,х);

Дальнейшая работа с Прологом-Д на **БК-0010** аналогична работе на «Ямахе» **MSX-2**.

Сравнение текстов по моделям позиций слов

Естественность диалога с компьютерной учебной программой, независимо от «направления» вопроса (от программы к обучаемому или наоборот), определяется способом анализа и оценки текста. Общепринятым подходом к анализу текста является выделение ключевых слов, представляющих фактическую сторону сообщения, и построение логических моделей распознавания текста. Например: «Анализ ответов осуществляется путем вычисления логического выражения, представленного в польской инверсной записи — ПОЛИЗ, где предикаты связаны операциями И, ИЛИ, НЕ. Определение значений предикатов производится по наличию (отсутствию) ключевых слов путем динамического маскирования текста ответа дескрипторами» [1].

Недостаток такого подхода — в неопределенности характера связи между дескриптивными, логическими терминами [2, с. 20—21] модели текста и проверяемым текстом. Логических операций И, ИЛИ, НЕ недостаточно для различения в естественном виде вводимого и заданного текстов. Поэтому необходимо не только раскрыть механизмы построения логических моделей, но и представить более эффективные средства анализа.

Предлагаемый способ анализа текста основывается на том, что расположение (позиция) некоторых слов в предложении не влияет на общий смысл, а изменение позиций других слов изменяет смысл текста. Таким образом, задавая относительно друг друга позиции отдельных слов и оставляя свободными для выбора (обучаемым) позиции других слов, удастся выполнить сравнение вводимого текста с хранящейся в памяти ЭВМ моделью позиций слов.

Числовые значения позиций слов определяются по расстоянию в знаках от начала введенного текста до последнего (или, по усмотрению программиста, первого) знакоместа прове-

ряемого слова. Множество возможных позиций слова в тексте выражается суждением с двумя переменными: «Слово X имеет позицию Y». Первая переменная получает свое значение при включении проверяемого слова в модель позиций слов. Значение второй переменной определяется позицией этого же слова в проверяемом тексте.

В компьютерном анализе текста слово не является словом в обычном понимании. Слово для машины — последовательность символов. Такие последовательности символов названы ключевыми словами, если, по мнению составителя контрольной работы, они должны восприниматься как наиболее существенные (например, с точки зрения проверки знаний) части учебного текста. Поэтому формализации подвергается не содержание текста, а множество допустимых вариантов размещения ключевых слов в тексте, которое выражается в общей форме модели позиций слов.

Анализ содержания самого текста (предполагаемый текст) выполняется составителем контрольной работы, и по результатам этого анализа за выделенными ключевыми словами закрепляются сигнальные символы, управляющие выбором алгоритмов сравнения позиции каждого проверяемого слова с позициями найденных в проверяемом тексте предыдущих ключевых слов.

Следующей особенностью предлагаемого способа является использование двух частей проверяемого слова в ключевом слове модели позиций — знаковой и сигнальной. Знаковая часть ключевого слова — это и есть выделенная составителем контрольной работы часть учебного текста: от символа до предложений, в зависимости от цели проверки. Сигнальная часть ключевого слова имеет фиксированную длину в знаках в знакоместах деля пакета учебных программ и состоит из нескольких символов, определяющих допустимые позиции слов в проверяемом тек-

сте. Таким образом, ключевое слово модели позиций отличается от такого же ключевого слова, или просто слова, проверяемого текста наличием сигнальной части.

Сигнальная часть отражает логические связи между суждениями относительно вариантов размещения (выделенных составителем контрольной работы) последовательностей символов в проверяемом тексте. Это позволяет связывать отдельные суждения о позиции слов в единую модель позиций. Для каждой модели позиций слов существует множество вариантов размещения слов в тексте, удовлетворяющих истинности сложных суждений после подстановки в простые суждения «Слово X имеет позицию Y» слов и позиций слов.

46 Хотя логические операции проверки текста относятся только к сравнению (больше, меньше) числовых значений позиций слов, для записи сигнальной части удобно использовать символы, обозначающие логические термины, соответствующие соединению суждений о допустимых позициях слов.

Если изменение позиций слов в тексте не приводит к изменению смысла текста, то ключевые слова в сигнальных частях имеют знак конъюнкции &. Такое объединение ключевых слов необходимо для установления фактической стороны текста без учета позиций слов. Знак конъюнкции может указывать и на позиционное присоединение к данному слову слова с другими сигнальными символами. Нужно учитывать и особенности использования в диалоге неполных предложений. Например, в контексте «вопрос—ответ» часть слов может включаться или не включаться в ответ, не вызывая изменения смысла текста. Поэтому первое ключевое слово модели позиций всегда имеет знак конъюнкции, который обозначает соединение данного суждения о существовании отдельных непроверяемых и не включаемых в модель позиций слов.

Если ключевое слово имеет возможность замены в тексте другим словом, то такие дополнительные слова располагаются в модели позиций за данным ключевым словом с записью в сигнальных частях знака дизъюнкции V.

При проверке достаточно найти в тексте одно слово из дизъюнктивной группы ключевых слов. Для уменьшения времени проверки порядок расположения ключевых слов со знаками дизъюнкции определяется оценкой вероятности их появления в проверяемом тексте. В последовательности ключевых слов со знаками дизъюнкции сохраняется функция управляющих сигналов первого ключевого слова из дизъюнктивной группы. Ключевые слова со знаками дизъюнкции позволяют выполнить разделение вариантов ответа в модели позиций для отдельных ключевых слов.

В модель позиций слов можно включать ключевые слова с отрицанием, которые обозначаются в сигнальных частях знаком «минус». Обычно это связано с поиском наиболее вероятных ошибок в проверяемом тексте. В случае, если последовательность некоторых слов строго определена, то такие ключевые слова модели позиций имеют знак «больше». При проверке по ключевому слову со знаком > числовое значение его позиции в проверяемом тексте должно быть больше числового значения позиции предыдущего ключевого слова. Знак «равно» используется для обозначения проверки соседствования двух ключевых слов. Их последовательность не имеет значения, но между ними не должно быть других слов.

В некоторых случаях возникает необходимость выделения отдельных групп, не имеющих одинаковых ключевых слов. Для их различения ключевые слова одной группы обозначаются знаком деления /, другой — апострофом. Выделение слов в группы выполняется по таким условиям: они должны присутствовать только в одной группе слов и среди них не должно быть слов из другой группы. Позиции групп в тексте произвольны. Позиции слов в группах определяются другими сигнальными символами. Проверка выполняется сравнением интервалов, образованных числовыми значениями позиций крайних ключевых слов каждой группы. Все следующие пары групп ключевых слов различаются в проверяемом тексте таким же образом.

Применяются и другие управляющие сигналы. Двоеточие задает поиск ключевого слова в тексте после найденного предыдущего. В отличие от сигнала $>$, текст до предыдущего ключевого слова не проверяется. Знак возведения в степень используется для задания поиска ключевого слова после найденного в тексте крайнего справа (имеющего наибольшее числовое значение позиции) ключевого слова из групп слов со знаками $/$, $'$. Такой прием позволяет, например, проверять знаки пунктуации. Причем интервал поиска ключевого слова обычно задается составителем контрольной работы. Во всех случаях задание интервалов поиска ключевых слов в проверяемом тексте сокращает время проверки и уменьшает вероятность ошибок сравнения.

Иногда возникает необходимость включения в модель позиций слов нескольких различных вариантов ответа. Разделение на дизъюнктивные модели позиций слов для одного ответа выполняется специальными ключевыми словами: «ИЛИ-» (без восстановления текста проверяемого ответа), «ИЛЬ-» (с восстановлением текста проверяемого ответа). Роль управляющих сигналов выполняют сочетания символов «И-» и «Ь-», оформленные в виде ключевых слов для удобства чтения в тексте модели позиций слов.

Управляющие сигналы закрепляются и за другими текстами контрольной работы. Например, если комментарий к ответу начинается со знака пробела (управляющий сигнал), то он выводится на экран только в случае ошибок в позициях слов проверяемого текста.

Сигнальные символы имеют фиксированные позиции и приоритеты в выполнении. При исполнении программы из модели позиций последовательно выделяются ключевые слова и происходит разделение каждого слова на знаковую часть (для поиска слов в тексте) и два сигнальных символа (для управления работой программы).

На программный анализ текста (при длине символьной переменной 255 символов для модели позиций и текста ответа) затрачивается около 2 мин. Компьютерный контроль знаний можно

рассматривать как способ тестирования обучаемых, в котором подразумеваются краткие ответы и ограниченное общее время выполнения контрольной работы. Поэтому чаще используются тексты меньшей длины и в среднем обработка текста одного ответа занимает 10—25 с. Чтобы избежать длительных пауз в работе, на экран перед началом проверки текста выводится информация для пользователя, например текст правильного ответа для сравнения, инструкция для дальнейших действий и т. п.

Каждое найденное в проверяемом тексте ключевое слово заменяется цепочкой одинаковых символов, например «XXXXXX», что исключает повторную идентификацию ключевых слов. Это позволяет при необходимости контролировать результаты выборки машиной ключевых слов из проверяемого текста. Его первоначальный вид можно восстановить в случае повторной обработки («ИЛЬ-»).

При подсчете ключевых слов не учитываются ключевые слова со знаками дизъюнкции и отрицания. Число слов (соответствующих ключевым словам модели позиций), найденных в проверяемом тексте, подсчитывается по следующему правилу: из групп ключевых слов с обозначением дизъюнкции учитывается только по одному слову и учитываются все остальные ключевые слова с вычитанием ключевых слов, обозначенных отрицанием. Если получается отрицательное число, то соответствующей переменной присваивается значение 0. При проверке по дизъюнктивным моделям берется первый результат, превосходящий минимум точности ответа, заданной составителем контрольной работы. За точность сравнения текстов принято отношение числа ключевых слов, найденных в проверяемом тексте без нарушений заданного расположения, к числу ключевых слов модели позиций. Коэффициент точности принимает значения от 0 до 1.

Избыточность ответа проверяется сопоставлением длины текста (в знаковых местах) ответа с длиной образцового ответа, умноженной на коэффициент точности. Поправочные коэффициенты сопоставления или отказ от такой

проверки определяются экспериментальным путем. Избыточные ответы комментируются замечаниями типа «Мне не все понятно в Вашем ответе».

После завершения анализа текста программа выводит на экран дополнительную учебную информацию, оценочные и мотивационные суждения. В случае оценки ответа как неверного студент может пригласить преподавателя и обосновать свой вариант ответа. Если преподаватель согласился с правильностью ответа, то студент за выполнение контрольной работы получает оценку «отлично», а программист включает в модель позиций слов дополнительные ключевые слова, дополнительные дизъюнктивные модели (или уточняется формулировка вопроса).

48

В качестве иллюстрации рассмотрим фрагменты компьютерных программ контрольных работ по курсу педагогики. Программы состоят из двух частей: текстов отдельных контрольных работ и общей для всех текстов программы проверки контрольных работ. Для упрощения из текстов контрольных работ приводятся только тексты контрольных вопросов, правильных ответов для сравнения, моделей позиций ключевых слов и по одному из возможных вариантов ответов студента, оцениваемых программой как точный ответ.

Вопрос 1. К какому типу шкал измерений относится общепринятая пятибалльная шкала оценок успеваемости школьников?

Ответ. Пятибалльная шкала оценок знаний, умений и навыков школьников относится к шкале порядка (ранжирования).

Модель.

"ПОРЯДОК № РАНЖИР В№ НАИМЕНОВ -№
ИНТЕРВАЛ -№ ОТНОШЕН -№#"

Ответ студента. Шкала оценок успеваемости школьников является шкалой ранжирования.

Вопрос 2. В виде соответствующей пропускам последовательности запишите для каждого случая названия противоположных или противоречивых мотивов учения: 1) ситуативные —; 2) устойчивые —; 3) положительные —

Ответ. В ответе можно назвать следующие мотивы: 1) перспективные; 2) неустойчивые; 3) отрицательные.

Модель.

"ПЕРСПЕК № НЕСИТУАТИВ В№ НЕУСТОЙЧ >№
ОТРИЦ >№ НЕПОЛОЖИТ В№ НЕЙТРАЛЬ -№
ПЕРСПЕК -№#"

Ответ студента. 1 — неситуативные, 2 — неустойчивые, 3 — отрицательные.

Вопрос 3. Какие уровни умственного развития школьников были выделены в концепции Л. С. Выготского?

Ответ. Выделены два уровня: первый — уровень актуального развития, второй — зона ближайшего развития школьника.

Модель.

"АКТУАЛЬН/ № ПЕРВ/ № 1/ В№. ! №. ! В№ТОР
' № 2' В№ БЛИЖ' №. ! №. ! В№УРОВ № ЗОН В№
РАЗВИТ №#"

Ответ студента. Уровень ближайшего развития — это второй уровень. Первый уровень — актуальное развитие ученика.

В моделях позиций ключевые слова отделены друг от друга символом \odot (он выбран потому, что не используется в знаковых частях ключевых слов контрольных работ). Символ $\#$ является сигналом завершения проверки одного ответа. Первая позиция за знаковой частью ключевого слова отводится для записи символов /, ', , : . Знак пробела — отказ от указанных выше управляющих сигналов. Вторая позиция предназначена для записи символов &, \vee , —, >, =. Все эти символы можно использовать для записи знаковой части ключевого слова. Пробел перед знаковой частью ключевого слова позволяет определить начало этого слова в проверяемом тексте.

Для уменьшения времени проверки удобно использовать части слов. При этом возрастает вероятность оценивания неточного ответа как правильного, но, поскольку обучаемый всегда сравнивает свой ответ с ответом программы, необходимое исправление (понимание неточностей) выполняется самим студентом.

Особое внимание при составлении текстов контрольных работ обращается на

благожелательность формулировок и разнообразие оценочных суждений. Не следует также стремиться к оценке каждого действия студента. Компьютерный контроль знаний как часть контроля обучения имеет целью не столько выяснение фактических знаний студентов, сколько организацию учебной деятельности.

Программы прошли экспериментальную проверку в преподавании курса педагогики в Башкирском государственном педагогическом институте в 1987—1989 гг. с использованием персональных ЭВМ «Искра-1256» и «Ямаха». Анкетирование студентов, слушателей ФППК, ИУУ показало: предварительные высокие оценки педагогической эффективности компьютерного контроля знаний после работы с программами

анализа текста имеют явно выраженную тенденцию к увеличению. К достоинствам программ анализа текста относятся и возможность составления и ввода программ текстов контрольных работ по различным дисциплинам преподавателями и студентами, имеющими самые элементарные навыки обращения с персональным компьютером.

Литература

1. Джалишвили З. О., Кириллов А. В. Принципы организации АОС по гуманитарным предметам // Материалы IV всесоюзного семинара «Разработка и применение программных средств ПЭВМ в учебном процессе». Тезисы докладов. Симферополь; М.: ИПИАН, 1988.

2. Гетманова А. Д. Логика: Учебник для студентов пед. вузов. М.: Высшая школа, 1986.

А. КАРАВАЕВ

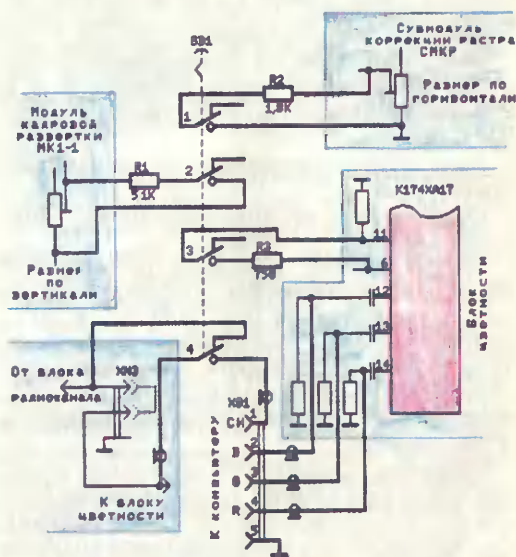
Цветной демонстрационный монитор

Учебный процесс пойдет гораздо эффективнее, если в дисплейном классе установить демонстрационный монитор с экраном 62 см по диагонали. Особенно он необходим в тех КУВТ, где используются черно-белые мониторы. Если телевизор еще не приобретен, то при его выборе необходимо обратить внимание, какие микросхемы используются в модуле цветности. Подключение облегчается, если в этом блоке используется микросхема K174XA17 (например, модули цветности МЦ-31, МЦ-1-5). В данной микросхеме имеются внутренние управляемые ключи, позволяющие превратить телевизор в цветной дисплей.

Описанная ниже схема разработана применительно к компьютерам комплекса УКНЦ, так как в этом случае коммутация получается более сложной. Причина в том, что при заданном формате изображение «не входит в экран». Поэтому приходится принимать меры для сжатия изображения по вертикали и горизонтали. На рис. 1 изображена принципиальная схема подключения цветного телевизора к одной из машин учащихся.

Черно-белый монитор при этом отключать не нужно, так как у компьютера имеется отдельный разъем для подключения цветного дисплея.

При отжатом переключателе SB1 телевизор работает в обычном режиме (прием телепрограмм), так как внутренние ключи микросхемы K174XA17 запрещают прохождение сигналов цветности (B, G, R), поступающих от компьютера через разъем XS1 на входы 12, 13, 14 микросхемы. В этом положении



сигнал от блока радиоканала проходит к блоку цветности через замкнутые контакты секции 4 переключателя SB1. Эти контакты заменяют перемычку XN3 (изображенную пунктиром), которую следует убрать.

Во втором положении переключателя (при нажатой кнопке) к модулю цветности будут поступать синхроимпульсы (СИ) через гнездо 1 разъема XS1. Через контакты секции 3 переключателя SB1 и резистор R3 с вывода 6 микросхемы поступает управляющее напряжение на ее вывод 11. При этом внутренние ключи разрешают прохождение сигналов с XS1 к видеоусилителям. Телевизор становится цветным дисплеем.

Для сжатия изображения по вертикали и горизонтали с помощью секций 1 и 2 переключателя подключаются шунтирующие

резисторы (R2 — к submodule коррекции раstra, R3 — к модулю кадровой развертки).

Переключатель SB1 установлен около антенного входа телевизора. Здесь же устанавливается разъем XS1 (например, 5-гнездовая розетка от магнитофонных входов). Сигналы от компьютера к телевизору поступают по 4-проводной линии, выполненной из тонкого телевизионного кабеля. Высокое качество цветного изображения получилось даже при длине линии 15 м.

Более толстыми линиями на схеме показаны добавленные к имеющимся в телевизоре новые элементы и соединения.

Схема подключения телевизора к КУВТ-86 проще. В этом случае секции 1 и 2 переключателя SB1 и вся связанная с ними коммутация будут лишними.

П. ХОДАКОВ

Спасение файлов на «Агате»

Все работающие на «Агате» рано или поздно (обычно рано) сталкиваются с ошибкой I/O ERROR, являющейся бичом всех пользователей «Агата». Не было бы ничего страшного, если бы эта «фатальная» ошибка поражала только те файлы, у которых есть дубликат. Ну а если...?

Спасению вашего файла, а заодно, может быть, и чужих, посвящается эта статья.

Чтобы пользоваться приведенными здесь рекомендациями, нужно:

- иметь машину «Агат»;
- желательно две;
- желательно еще больше;
- иметь программу «Агат-отладчик»;
- уметь работать с ней в режимах К и I;
- иметь понятие о структуре (треках и секторах) диска;
- знать шестнадцатиричную систему счисления;
- иметь «штатную» (поставляемую с «Агатом») систему.

1. Что делать, если файл не читается? Прежде всего попробуйте прочитать его на той машине, где он был записан. В 50 % случаев это дает положительный эффект. Если такой возможности нет, все равно попробуйте прочитать его на других машинах. Если это не помогло, придется прибегнуть к «терапии».

«Жесткая терапия» дозволительна в случае, когда не жалко ни дисковод, ни дискету, а жалко только свой файл (т. е. самого себя). Она заключается в так называемой подкачке

дисковода: в момент потрескивания дисковода вы беретесь рукой за крышку, немного приоткрываете ее и закрываете, снова приоткрываете и закрываете — и так до тех пор, пока не прочитаете файл или снова не получите I/O ERROR. Иногда помогает просто сильное надавливание на крышку. Успех «жесткой терапии» в основном зависит от опыта «терапевта».

Для использования «мягкой терапии» прежде всего нужно установить характер ошибки.

Случай 1. При попытке чтения каталога на экран ничего не выдается, а сразу после потрескивания появляется I/O ERROR. Диагноз ясен, болезнь излечима в 90 % случаев, но, увы, с побочными эффектами. Дефект притаился в секторе 0 трека 11. На нем находится основная информация о диске, используемая всеми командами ДОСа.

Лечение производится с помощью системной программы «Агат-отладчик» в режиме трек-секторного редактора. Необходимо взять «здоровый» диск (каталог которого читается) и считать с него сектор 0 трека 1Г. Затем вытащить этот диск, вставить «больной» и записать на него то, что вы считали со «здорового». Если запись прошла успешно, можете прыгать от счастья: все, что было на вашем диске, будет читаться. Но будущее не совсем безоблачно: на этот диск нельзя больше писать. Запись будет проходить нормально, но в один прекрасный момент вы можете лишиться своих спасенных файлов.

Чтобы этого не случилось, скопируйте все нужные вам файлы на другой диск, затем отформатируйте спасенный и после этого перепишите файлы обратно.

Если запись на сектор 0 трека 11 не прошла, можно попробовать скопировать весь диск целиком на другой и уже на дубликате произвести указанное выше вмешательство.

Случай 2. После команды CATALOG появилась надпись DISK VOLUME..., а возможно, и часть каталога (в которой вашего файла нет), и уже потом I/O ERROR. Время покажет, можно ли восстановить ваш файл. Диагноз же таков: запрещен один из блоков каталога, который располагается все на том же треке 11, начиная с сектора F (имеется в виду порядок вывода на экран) и далее E, D, C, B, A, 9, 8, 7, 6, 5, 4, 3, 2, 1. Какой-то из них не читается.

Опять прибегнем к трек-секторному редактору. Будем поочередно прочитывать все сектора каталога и в конце концов найдем поврежденный.

Теперь займемся собственно «реанимацией». Вернемся на один сектор назад, к началу каталога, т. е. передвинемся с плохого сектора на хороший. Не забудьте, что за начало считается не 0, а F! Поставим курсор так, чтобы «позиция в секторе» была 1. Там будет число 11. Передвинем курсор в позицию 2. Там будет число, обозначающее номер следующего (в нашем случае нечитаемого) сектора. Изменим это число на номер первого хорошего сектора, следующего за плохим (тем самым мы велим машине обойти плохой сектор). Затем записываем только что измененный сектор на диск. Если на 11-м треке оказалось несколько плохих секторов, то поступаем с ними аналогично.

Снова выводим на экран каталог и ищем нужный нам файл. Если его нет — увы, ему пришел конец. Впрочем, можно еще вернуть диск в исходное состояние и попробовать «жесткую терапию».

Особый случай — если на экран выводится только DISK VOLUME... и I/O ERROR. Значит, запортился начальный сектор каталога — F. Обходим его точно так же, только ставим номер следующего после F читаемого сектора в позицию 2 сектора 0 трека 11.

Случай 3. Каталог читается, ваш файл в нем присутствует. Но на команду чтения файла ответ — I/O ERROR. Следовательно, запортился один из блоков, на которых записан ваш файл. К сожалению, спасти его в первоизданном виде не удастся, но большую часть файла, пожертвовав несколькими блоками, можно сохранить. Просмотрите в трек-секторном редакторе сектора каталога и найдите имя вашего вожденного нечитаемого файла. В каждом секторе каталога

умещается запись о семи файлах. Запомните, каким по очереди (сверху) стоит это имя, а затем поставьте курсор в соответствие с таблицей.

| Позиция имени файла | «Позиция в секторе» |
|---------------------|---------------------|
| 1 | B |
| 2 | 2E |
| 3 | 51 |
| 4 | 74 |
| 5 | 97 |
| 6 | BA |
| 7 | |

Запомните число, на котором находится курсор, а также число, стоящее правее. Первое — номер трека, второе — номер сектора «трек-секторного» списка, в котором хранится вся информация о местонахождении вашего файла на диске. Прочитав этот блок, вы увидите список блоков файла. Если длина файла больше 122 блоков, то у «трек-секторного» списка есть продолжение. Номера трека и сектора продолжения вы найдете в позициях 1 и 2 соответственно.

Если вы не сможете прочитать этот блок — что ж, есть еще «жесткая терапия». В нормальной же ситуации поставьте курсор в позицию C. Спишите или запомните по парам все числа. Как вы уже догадались, первое — номер трека, второе — сектора каждого блока вашего файла. Вам предстоит большая работа. Просмотрев все блоки, найдите нечитаемые. Их может быть несколько. Запомните их, а затем возвратитесь в «трек-секторный» список. В нем, начиная с позиции C, найдите пару чисел, показывающих трек и сектор плохого блока. Замените их. На что? Да на трек и сектор блока, который стоит рядом — пусть считается два раза, а уже потом вы легко сможете удалить лишнее. Не забудьте записать измененный список. Теперь осталось только загрузить ваш файл, удалить лишнее, добавить недостающее.

Теперь разберем ситуации, когда файл не удастся записать. Проще всего тогда сменить диск. А если другого нет? Работать в этом случае придется на двух машинах. Пока у одной в памяти находится ваш файл, на другой готовится диск.

Средство 1. Загрузить «Агат-отладчик» и в режиме копирования дисков скопировать диск сам на себя. Тут надо быть осторожным. Иногда некоторые треки читаются, а затем не пишутся — плохой дисковод. Да еще впоследствии этот диск не форматруется. Но даже если перезапись прошла успешно, программы на Бейсике могут неузнаваемо искажаться. С текстовыми и бинарными файлами такого замечено не

было, но надежнее пользоваться другими средствами.

Средство 2. В трек-секторном редакторе читаете сектор 0 трека 11. Ставите курсор в 30-ю позицию. Число, на котором стоит курсор,— номер трека выделения: начиная с этого трека будет при записи выделяться место для хранения вашего файла. А раз ваш файл не пишется, то где-то на этом треке есть плохой блок, через который нужно «перепрыгнуть». Для этого поставьте в позицию 30 другой номер трека, например на 1 больше, и запишите измененный сектор. После этого можно снова попробовать записывать на этот диск.

Средство 3. Его хорошо использовать вместе со средством номер 2. Суть состоит в запрещении использования сбойного трека. С помощью режима копирования в «Агат-отладчике» (когда машина только читает диск) определите, какие треки плохие, и запишите их. Прочтите сектор 0 трека 11. Назначение этого блока — показывать, что на диске занято, а что свободно. В нем каждому треку отведено четыре числа. Два из них ничего не значат (в них всегда 0), а в двух других закодирована некая информация. Для запрещения записи на плохой трек достаточно поставить в эти две позиции нули. Выполнив это, не забудьте записать измененный сектор. Теперь при записи машина будет обходить плохие треки. Определить, в какую позицию заносить нули, поможет вам таблица.

| Номер плохой трека | Позиция для занесения нулей |
|--------------------------|-----------------------------------|
| 0 | 38—39 |
| 1 | 3С—3D |
| 2 | 40—41 |
| 3 | 44—45 |
| 4 | 48—49 |
| 5 | 4С—4D |
| 6 | 50—51 |
| 7 | 54—55 |
| 8 | 58—59 |
| 9 | 5С—5D |
| A | 60—61 |
| B | 64—65 |
| C | 68—69 |
| D | 6С—6D |
| E | 70—71 |
| F | 74—75 |
| 10 | 78—79 |
| 11 | 7С—7D |
| 12 | 80—81 |
| 13 | 84—85 |
| 14 | 88—89 |
| 15 | 8С—8D |
| 16 | 90—91 |
| 17 | 94—95 |
| 18 | 98—99 |
| 19 | 9С—9D |
| 1A | A0—A1 |
| 1B | A4—A5 |
| 1C | A8—A9 |
| 1D | AC—AD |
| 1E | B0—B1 |
| 1F | B4—B5 |
| 20 | B8—B9 |
| 21 | BC—BD |
| 22 | CO—C1 |

52

Последняя тема статьи — нештатные ситуации при создании программ на Бейсике. Общеизвестно, что «Агат» — машина загадочная и капризная. Не нужно долго работать на ней, чтобы попасть в ситуацию, когда программа почему-то не записывается. Для этого случая несколько советов.

Совет 1. Никогда не создавайте программы в SUPER DOS, поскольку это очень капризная система.

Совет 2. Если на команду CATALOG, а также на все остальные команды ДОСа система выдает SINTAX ERROR, попробуйте набрать: CALL X 9DBF <ВВОД>. Снова наберите CATALOG. Если дисковод заработал, смело записывайте программу, но не нажимайте СБР УПР: снова произойдет сбой.

Совет 3. Нередки случаи, когда между номерами строк забивается какой-то номер и никакими командами стирания (DEL) его не удастся уничтожить. В Бейсике для таких

ситуаций есть специальная функция FRE(X) (набирается ?FRE(0)), но она малоэффективна. Если она вам не поможет, попробуйте превратить бейсиковый файл в текстовый. Это делается при помощи небольшой программки, которая располагается так, чтобы не забивать вашу основную программу. Пример: пусть ваша программа занимает номера с 1 по 40 000. Добавляете к ней такие строки:

```
60000 INPUT "ВВЕДИТЕ ИМЯ ФАЙЛА":NR
60010 ?:? CHR# (4): "OPEN": NR
60020 ?CHR# (4): "WRITE": NR
60030 LIST 1. 40000
60040 ?CHR# (4): "CLOSE": NR
60050 END
```

На RUN 60 000 машина запрашивает имя файла. После ввода имени диск закрутится, и после окончания его работы в каталоге вы найдете текстовый файл с введенным вами именем. Это текстовый вариант основной программы. С помощью «Школьницы» или «Агат-ассемблера» считайте этот файл и в редакторе удалите строптивный номер. Записав

файл на место, загрузите Бейсик и дайте команду EXEC <имя файла>, а затем LIST. Перед вами программа без противного номера.

Вот и все. В заключение — пара полезных истин.

Истина 1. На каждом диске 35=0—

— X 22 трека, а в каждом треке 16=0—F секторов.

Истина 2. То, что написано в данной статье, конечно же, неполно. Существует еще много способов восстановления файлов, но, чтобы ими пользоваться, необходимо иметь специальную подготовку.

Р. САБИРОВ

Хива

Простой клавиатурный тренажер

По мере распространения учебной ВТ все менее терпимой становится парадоксальная ситуация, когда КУВТ простаивает из-за отсутствия ППС. Увы, такое еще встречается. Предлагаю вниманию читателей «ИНФО» простую программу — клавиатурный тренажер «Быстрая рука». Она доказала на практике свою высокую эффективность, а кроме того, может послужить примером простых, но практически полезных программ, создавать которые на факультативных занятиях учащихся гораздо интереснее, чем разбирать абстрактные примеры.

```
5 'БЫСТРАЯ РУКА
100 KEYOFF:COLOR 8,3:CLS:WIDTH40
110 LOCATE 0,12:PRINT"
<<<<<БЫСТРАЯ РУКА>>>>>"
120 LOCATE 7,14:
PRINT"[Клавиатурный тренажер]"
130 LOCATE 0,20:
PRINT"Авторы программы:"
140 LOCATE 2,22:
PRINT"Р.И.Сабиров, М.А.Абдурасулов"
150 LOCATE 10,23:
PRINT"[город Хива, 1989]"
160 LOCATE 8,33:PRINT"<нажмите пробел>"
170 IF INKEY#=CHR$(32) THEN 171
ELSE 170
171 CLS: LOCATE 14,6:PRINT"РЕЖИМ"
172 LOCATE 8,10:
PRINT"клавиша [CAPS] включена"
173 LOCATE 8,12:
PRINT"клавиша [ПУС] отключена"
175 IF INKEY#=CHR$(32) THEN 200
ELSE 172
200 CLS:COLOR10,1:S=0:V=0:G=.4
210 R=RND(-TIME)
220 V=V+1
230 N=INT(RND(R)*25)+65
240 FOR I=1 TO 36 STEP G
250 LOCATE I,12:PRINT CHR$(N)
260 A$=INKEY$
270 IF V=6 THEN 320
280 IF A#=CHR$(N) THEN 300
290 CLS:NEXT I:GOTO 210
300 PLAY"m65000S8;N74;64":S=S+1
310 GOTO 210
320 H=H+1:V=0:G=G+.1
330 IF H=6 THEN 335 ELSE 210
335 LOCATE 10,8:
```

```
PRINT"РЕЗУЛЬТАТ"
340 LOCATE 0,12:PRINT
"ПОПЫТОК - 30", "ПОЙМАНО - "S
341 PRINT:PRINT:PRINT
"ВАША РЕАКЦИЯ РАВНА"INT(S/30*100)"%
ОТ ИДЕАЛЬНОЙ"
342 FOR Z=0 TO 2500:NEXT Z
345 CLS
350 LOCATE 0,12:
PRINT"ХОТИТЕ ПОВТОРИТЬ?"
360 LOCATE 18,12:PRINT"[ДА-1/НЕТ-0]"
370 K$=INPUT$(1)
375 IF K$="1" THEN RUN
380 CLS: LOCATE 14,10:
PRINT"ДО СВИДАНИЯ!"
390 LOCATE 12,12:PRINT"ВСЕГО ХОРОШЕГО!"
400 END
```

53

После запуска программы с левого края экрана на правый начинают пробегать произвольно выбираемые ЭВМ буквы. Ученик должен вовремя «поймать» их, т. е. нажать на соответствующую клавишу. Вовремя «пойманная» буква отзывается звуковым сигналом и приплюсовывается к общему результату. Скорость пробега увеличивается после каждых пяти букв. После пробега тридцати букв на экране высвечиваются результаты упражнения и реакция ученика в процентах к идеальной.

В данном варианте программа предназначена для упражнений с прописными латинскими буквами. Она легко «перенастраивается» для упражнений со строчными латинскими прописными и строчными русскими буквами, цифрами и спецсимволами одним из приводимых ниже изменений строки 230:

```
230 N=INT(RND(1)*26)+97
230 N=INT(RND(1)*32)+224
230 N=INT(RND(1)*32)+192
230 N=INT(RND(1)*32)+33
```

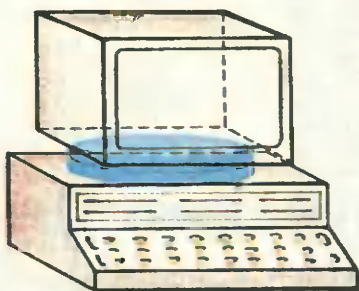
Если же дать строку 230 в виде

```
230 N=INT(RND(1)*223)+33
```

то получается тренажер с очень сложной подборкой символов по всем группам.

Неожиданное применение

При отсутствии в КВТ (у нас в нем стоят «Ямахи») цветного демонстрационного монитора его роль может не без успеха сыграть монитор РМП. Для этого нужно придать ему «маневренность» — например, с помощью... диска-тренажера «Грация». Последний предназначен для домашних занятий физкультурой и состоит из двух дисков, вращающихся относительно друг друга. Установив на «Грацию» монитор РМП и дав определенную свободу проводам, учитель может легко поворачивать его экраном к классу.



Программа печатает сама себя (Компьютерный аукцион)

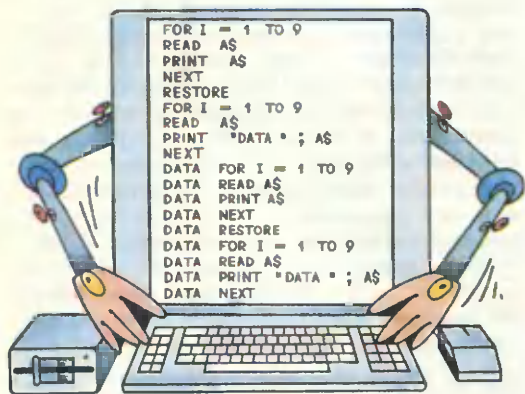
Нет, наверно, в мире компьютерного журнала, который бы не осветил в той или иной степени проблему самопечати программ. Наш журнал не хочет отставать в этом отношении. Мы объявляем конкурс — аукцион «наоборот»: на обычном аукционе ведущий выкрикивает «Кто больше?», а на нашем компьютерном аукционе ведущий будет выкрикивать «Кто меньше?» — «Кто напишет Бейсик-про-

грамму самопечати с меньшим числом операторов?» Начальная ставка аукциона — программа 1 на Квик-Бейсике (Турбо-Бейсике), составленная А. Сибиряковым (г. Канск) и оформленная (единство формы и содержания!) Н. Сушенцовым (г. Киров).

Предупреждаем! Программы типа 1 LIST 1 рассматриваться не будут. Во-первых, оператора (команды) LIST нет во многих версиях Бейсика. Во-вторых, и это главное, программа 1 LIST 1 не предел. Последнюю ставку нашего аукциона нетрудно предугадать. Это программа с нулем операторов. Но нас интересует не итог, а сам процесс, т. е. промежуточные программы, а главное, идеи, которые наши читатели заложат в них.

Третий удар молотка, подытоживающий компьютерный аукцион, мы услышим в конце 1991 г.

Итак, начальная ставка названа, аукцион начался. Мы готовы вести подобные аукционы и по другим языкам программирования, если читатели пришлют нам программы самопечати, написанные на Паскале, Си, Фортране и др. Ждем также идей новых аукционов и конкурсов.



Поправка. В статье А. Свердлова «Моделирование многомерных процессов» (ИНФО. 1990. № 2) в третьей строке правого столбца на с. 85 должно стоять не N, а n; на с. 86 в последней строке левого столбца выражение в фигурных скобках следует читать $a_{ijk}(i_0+h)$.

Как предупредить переутомление при работе с видеотерминалом?

Ранее проведенными гигиеническими исследованиями на студентах [1, 2, 3 и др.], а в дальнейшем — и на школьниках [4—8 и др.] было показано, что работа на видеотерминалах вызывает значительное напряжение центральной нервной системы, зрительного анализатора, большая нагрузка приходится на сердечно-сосудистую и мышечную системы, связана с гиподинамией. Поэтому очень важно разработать комплексы мероприятий, способствующие предупреждению развития утомления и переутомления.

Целью исследований было разработать комплексы упражнений для глаз, направленные на снижение утомляемости зрительного анализатора, апробировать в процессе работы студентов за видеотерминалами комплексы упражнений физкультурных пауз (ФП), разработанных НИИ физической культуры [9].

Комплексы упражнений для глаз были составлены нами на основании данных литературы. Эти комплексы утверждены Минздравом СССР так же, как приведенные ФП [10].

Исследования проводились в учебном помещении, оборудованном девятью персональными электронно-вычислительными машинами (ПЭВМ) «Ямаха MSX-1» на базовом объекте МИФИ.

Расположение рабочих мест с ПЭВМ периметральное, по двум сторонам. Работа за видеотерминалами осуществлялась при общем люминесцентном освещении (лампы ЛБ). Венгерские светильники с экранящими решетками были вмонтированы в подшивной звукопоглощающий потолок. При включении осветительных установок исключалась блескость на экранах мониторов. На рабочей поверхности стола создавалась хорошая освещенность порядка 480—400 лк, на экране монитора — 280—300 лк. Во время работы осуществлялась широкая аэрация через открытые зашторенные окна.

Под наблюдением находились девять студентов (студенческий отряд), закончивших третий курс. За каждым студентом закреплялось рабочее место с ПЭВМ. Студенты работали пять дней в неделю, средняя длительность работы в течение дня — шесть часов. В середине рабочего дня устраивался перерыв в один час для обеда и отдыха. Студенты разрабатывали учебные программы

по разным предметам непосредственно на ПЭВМ. Плотность работы на ПЭВМ была высокой и составляла 95—97 %.

Работа осуществлялась при оптимальном сочетании цветов информации и фона экрана: темно-зеленый фон и белые знаки [6]. При выполнении работ на ПЭВМ следили за правильной посадкой и оптимальным расстоянием глаз от экрана монитора (60—70 см).

Изучалось функциональное состояние центральной нервной системы, зрительного анализатора и сердечно-сосудистой системы четыре раза: до начала работы, до и после обеденного перерыва и в конце рабочего дня. Первую неделю студенты работали без осуществления профилактических мероприятий (контроль), последующие недели с их проведением.

Режим работы с введением профилактических мероприятий строился следующим образом: через 20—25 мин работы осуществлялся комплекс упражнений для глаз (изучены три варианта комплексов упражнений для глаз), через 50 мин работы — ФП, при этом упражнения сочетались с движением глаз.

Обработка результатов физиологических исследований осуществлялась методом аддитивно-нелинейного описания на ЭВМ [11] в зависимости от времени работы (до и после обеда), дней недели, вариантов упражнений для глаз, повторности их проведения, а также проведения ФП.

Наихудшие результаты по показателям центральной нервной, сердечно-сосудистой систем и зрительного анализатора были получены, когда студенты работали за ПЭВМ без проведения упражнений для глаз и ФП, при этом отмечались жалобы на общую усталость, усталость глаз и головную боль.

Напряженная умственная работа за видеотерминалом, пониженная двигательная активность, как показали ранее проведенные исследования [2] и настоящие, отрицательно сказывались на функциональном состоянии сердечно-сосудистой системы, главным образом артериальном давлении, являющемся интегральным показателем, особенно у лиц, не тренированных к физическим нагрузкам. Артериальное давление повышалось, снижался ударный объем крови (количество

крови, выбрасываемое в русло сосудов за одно сокращение сердца), который определяет работу всей системы кровообращения [12] почти при неизменной частоте сердечного ритма. Наряду со снижением ударного объема крови уменьшается минутный объем крови, происходят застойные явления в нижней половине тела и ног, ухудшаются окислительно-восстановительные процессы, особенно в центральной нервной системе, что в свою очередь способствует быстрому наступлению утомления и переутомления, появлению головной боли.

На основании проведенных исследований можно сделать следующие выводы.

Введение комплексов упражнений для глаз способствовало снижению утомления зрительного анализатора. Положительная динамика функционального состояния зрительного анализатора по всем изучаемым показателям (хронорефлексометрия КЧССМ, объем аккомодации, коэффициент утомляемости) наблюдалась при выполнении всех трех комплексов упражнений для глаз [10]. Кроме того, применение комплекса упражнений для глаз приводило к некоторому улучшению функционального состояния центральной нервной и сердечно-сосудистой системы.

Упражнения для глаз необходимо проводить при работе на ПЭВМ «Ямаха MSX-1» через 20—25 мин.

Выполнение комплекса упражнений ФП способствовало улучшению функционального состояния сердечно-сосудистой системы студентов, работающих на ПЭВМ «Ямаха MSX-1».

ФП следует проводить через 45—50 мин работы за видеотерминалами.

Изучение динамики функционального состояния центральной нервной и сердечно-сосудистой систем в течение недели и всего периода работы студенческого отряда (1 мес) не позволяет положительно оценить шестичасовой режим работы за видеотерминалами, несмотря на введение профилактических мероприятий.

Ежедневная длительность работы студентов за видеотерминалами, как показали исследования функционального состояния, во

вне учебное время (студенческий отряд) не должна превышать трех часов.

Литература

1. Гельтищева Е. А., Орлова Т. В., Циркова Н. Л., Шабалина Т. А., Яковлев Л. Т., Шилкин А. А., Гайнулина Г. М. Гигиеническая оценка условий работы и функционального состояния студентов при работе в дисплейном классе // Актуальные проблемы медицинского обеспечения студентов и учащихся: Сб. М., 1985. С. 43—44.
2. Гельтищева Е. А. Функциональное состояние сердечно-сосудистой системы студентов технического вуза при работе с дисплеями ЭВМ во время учебных занятий // Гигиена труда и профессиональные заболевания. 1985. № 6. С. 50—51.
3. Шицкова А. П., Ястребов Г. Г., Гельтищева Е. А. Гигиенические проблемы применения технических средств обучения в вузах и школах // Гигиена и санитария. 1987. № 6. С. 4—7.
4. Гельтищева Е. А. Некоторые вопросы профилактики утомления при работе на ЭВМ в школе // Вопросы психологии. 1986. № 5. С. 88—90.
5. Гельтищева Е. А., Селехова Г. Режим работы за дисплеем // Информатика и образование. 1987. № 1. С. 82—84.
6. Гельтищева Е., Селехова Г. Гигиеническое обоснование сочетания цветов на видеотерминалах // Информатика и образование. 1989. № 1. С. 74—78.
7. Мордвинов А. Г. Гигиеническое обоснование режима учебных занятий по основам информатики и вычислительной техники в старших классах с использованием компьютеров // Гигиена и санитария. 1989. № 9. С. 8—10.
8. Леонова Л. А., Савватеева С. С., Митрохина И. В. Влияние работы на дисплее на состояние зрительного анализатора учащихся // Гигиенические проблемы компьютеризации общеобразовательной школы. Сб. тр. М., 1988. С. 42—48.
9. Косилкина Н. К., Колгановский А. П. Производственная гимнастика для работников умственного труда. М.: Физкультура и спорт. 1983.
10. Методические указания по профилактике переутомления студентов вузов при работе с видеотерминалами, утв. Минздравом СССР 05.03.1988 № 4563—88 и согласованные с Минвузом СССР 09.03.88. № 96—21—05/65.
11. Указания по применению симметричных и несимметричных равномерных и неравномерных планов для нахождения оптимальных условий функционирования объектов медицинской промышленности. М.: ЦБ АСУ. 1981. 55 с.
12. Тулицын И. О. Возрастная динамика и адаптационные изменения сердечно-сосудистой системы школьников. М., 1985.

Дополнение. Для расширения памяти «Микроши» (статья «Микроша: 48К байт», «ИНФО» № 3) используются конденсаторы емкостью 0,1 мкФ; сопротивление R1 — 1 кОм, остальные сопротивления по 33 ома; микросхемы D1, D2 — K555КП11, D3 — K155ИР1, D4 — K155ЛА3, D5 — K555ЛЕ1.

О доплате за заведование кабинетом вычислительной техники и за работающие компьютеры

В связи с поступающими запросами о применении п. 66 Инструкции о порядке исчисления заработной платы работников просвещения, утвержденной приказом Министерства просвещения СССР от 16 мая 1985 года № 94, и постановления Государственного комитета СССР по труду и социальным вопросам и Секретариата ВЦСПС от 26 июля 1985 года № 242/16—42 «Об условиях привлечения квалифицированных специалистов и преподавателей по вычислительной технике и программированию вузов, научно-исследовательских институтов и предприятий для преподавания курса «Основы информатики и вычислительной техники» Министерства просвещения СССР по согласованию с управлением финансирования культуры и здравоохранения Минфина СССР, отделом социально-культурных отраслей Госкомтруда СССР и отделом заработной платы и экономической работы ВЦСПС сообщает следующее.

Доплата учителям школ за заведование кабинетом вычислительной техники (ВТ) в размере 10 руб. учитывается при тарификации на начало учебного года.

Заведование кабинетом ВТ в школе возлагается приказом директора на учителя, преподающего предмет «Основы информатики и вычислительной техники» («ОИ и ВТ»), который обеспечивает использование кабинета в соответствии с учебной программой, принимает меры по его оборудованию и пополнению учебно-наглядными пособиями и техническими средствами обучения, несет ответственность за сохранность имеющегося в кабинете оборудования.

Кроме того, учителю школы, на которого возложено заведование кабинетом ВТ, производится доплата в размере 5 руб. за каждый работающий компьютер.

При этом он обязан осуществлять профилактическое обслуживание компьютеров согласно инструкции по эксплуатации, проверку их технического состояния, а в случае неисправности вверенного ему оборудования организовывать ремонт представителями предприятий «Союзсметтехники» в соответствии с заключенным договором.

Общий размер доплаты за работающие на каждом рабочем месте ученика и учителя персональные компьютеры или терминалы с алфавитно-цифровой клавиатурой и устройством отображения информации (дисплеем), подключенные к ЭВМ, определяется еже-

месячно в зависимости от количества исправных компьютеров или терминалов и при тарификации учителей не учитывается. За микрокалькуляторы действующим законодательством дополнительной оплаты не установлено.

Сведения о количестве работающих компьютеров или терминалов и размере доплаты подаются одновременно с представлением табеля для начисления заработной платы работникам школы согласно прилагаемой форме. Исправность вычислительной техники определяется один раз в месяц, ко дню подачи табеля комиссией в составе директора школы или его заместителя, члена профкома и учителя, преподающего учебный предмет «ОИ и ВТ».

Следует иметь в виду, что кабинеты ВТ могут создаваться для одной школы или оборудоваться для нескольких школ как межшкольные. При этом межшкольный кабинет ВТ может создаваться в здании одной из школ или размещаться в другом, специально отведенном для этого помещении. Межшкольный кабинет ВТ является учебным кабинетом одной из школ и показывается в статотчетности данной школы с пометкой «межшкольный».

Доплаты в указанном выше порядке производятся за кабинеты ВТ, оборудованные органами народного образования (в том числе с участием шефствующих предприятий, учреждений и организаций) для проведения занятий с учащимися по курсу «ОИ и ВТ» и по другим общеобразовательным предметам.

Учебный предмет «ОИ и ВТ», как правило, должен вести учитель школы, прошедший специальную подготовку для преподавания этого курса. При проведении занятий по данному предмету с учащимися разных школ оплата труда учителя, для которого работа в школе является основной, производится на условиях, предусмотренных для учителей, работающих в нескольких школах.

Если к преподаванию «ОИ и ВТ» привлекаются специалисты предприятий, учреждений и организаций, то на них постановлением Госкомтруда СССР и Секретариата ВЦСПС от 26 июля 1985 г. № 242/16—42 распространены условия штатного совместительства. Это дает основание руководителю школы возлагать на них обязанности по заведованию кабинетом ВТ и обслуживанию вычислительной техники с соответствующей

доплатой. За преподавательскую работу к указанным лицам следует применять условия почасовой оплаты.

Привлекая для преподавания указанного предмета специалистов предприятий, необходимо соблюдать следующее: совместительство может иметь место только в одном учреждении (в одной школе), оплата за преподавательскую работу с учетом доплат у этих работников не должна превышать размера, соответствующего половине их месячной ставки заработной платы, установленной в соответствии с педагогическим стажем и образованием. В каникулярное время выплата заработной платы указанным учителям не производится.

За кабинеты ВТ, созданные предприятиями, учреждениями, организациями в УПК для трудовой (профессиональной) подготовки учащихся или во внешкольных учреждениях для развития детского технического творчества, а также кабинеты, созданные

организациями народного образования (в том числе с помощью предприятий) в ИУУ для подготовки и переподготовки учителей и одновременно используемые для изучения школьного предмета «ОИ и ВТ», дополнительная оплата за заведование кабинетом и за работающие компьютеры действующим законодательством не установлена.

В институтах усовершенствования учителей в кабинетах ВТ на общих основаниях устанавливается должность заведующего, а в соответствии с приказом Министерства просвещения СССР от 13 декабря 1985 г. № 220 дополнительно введена должность методиста. Оплата труда этих работников осуществляется в порядке, предусмотренном для соответствующих категорий работников ИУУ.

Порядок и условия, предусмотренные настоящим письмом, следует применять так же к преподавателям педучилищ, ведущим курс «ОИ и ВТ».

58

Возвращаясь к теме...

В редакцию журнала «Информатика и образование» поступает большое количество писем, касающихся оплаты труда, режима работы, исчисления педагогического стажа преподавателям информатики и вычислительной техники, работающих в школах, профессионально-технических училищах и средних специальных учебных заведениях.

Редакция ознакомилась с указанными письмами работников Главного планово-экономического управления Гособразования СССР.

Как отмечается в комментарии заместителя начальника отдела финансирования, труда и заработной платы учреждений просвещения т. Б. В. Вилкова, письма свидетельствуют о том, что на местах руководители учебных заведений, органы народного образования в ряде случаев нарушают порядок оплаты труда работников, определенный постановлением ЦК КПСС, Совета Министров СССР № 271.

Еще в 1986 г. Министерством просвещения СССР было издано инструктивное письмо в адрес органов народного образования от 07.07.86 г. № 63-М «О доплате за заведование кабинетом вычислительной техники и за работающие компьютеры». В письме были даны соответствующие разъяснения, в том числе и на вопросы, содержащиеся в письмах в адрес журнала. Инструктивное письмо было опубликовано в Бюллетене нормативных

актов Министерства просвещения СССР (№ 10 за 1986 г.), а также во втором номере нашего журнала за 1986 г.

Не имея возможности ответить каждому автору письма, редакция решила еще раз опубликовать содержание Инструктивного письма бывшего Министерства просвещения СССР, которое продолжает действовать, и надеется, что при внимательном чтении все, у кого возникли вопросы, найдут на них ответ.

Свой комментарий к письмам преподавателей ПТУ дал заместитель начальника отдела финансирования и оплаты труда учреждений профтехобразования Л. К. Иванов.

В соответствии с Инструкцией о порядке исчисления заработной платы работников профессионально-технических учебных заведений (п. 42) преподавателям профессионально-технических учебных заведений, на которых возложено заведование учебными кабинетами по курсу «Основы информатики и вычислительной техники», производится дополнительная оплата в размере 10 руб. в месяц.

В учебных кабинетах, оснащенных электронно-вычислительными машинами установленных образцов, кроме того, на основании справок учебной части об их использовании в учебном процессе производится

доплата указанным преподавателям — ведущим кабинетами в размере 5 руб. в месяц за каждый работающий компьютер.

При составлении справок учебной части об использовании компьютеров в учебном процессе для исчисления размеров доплаты преподавателям — ведущим учебными кабинетами по курсу «Основы информатики и вычислительной техники» в размере 5 руб. в месяц за каждый работающий компьютер необходимо учитывать внеурочную работу на ЭВМ, работы по созданию программного обеспечения для использования в учебно-воспитательном процессе и управлении профессионально-техническими учебными заведениями, ведение занятий по другим дисциплинам с использованием компьютеров, т. е. под «работающим компьютером» необходимо понимать технически исправный и эксплуатируемый в указанных целях компьютер.

Например, учебные кабинеты оснащаются комплектами вычислительной техники «Электроника КУВТ-86», включающие персональные ЭВМ «Электроника БК-0010Ш» и ДВК-2М. Оплата по 5 руб. за каждый работающий компьютер должна производиться преподавателю, назначенному ведущим кабинетом, оснащенным указанной техникой.

Действующим законодательством преподаватели и ведущие учебными кабинетами учебных заведений не отнесены к числу

работников, с которыми могут заключаться письменные договоры о полной материальной ответственности.

Дисциплина «Основы информатики и вычислительной техники» в ПТУ отнесена к числу специальных, что позволяет засчитывать в стаж педагогической работы преподавателей этих дисциплин время работы на предприятиях, в учреждениях и организациях в порядке, предусмотренном для преподавателей специальных дисциплин, т. е. при условии работы в течение не менее 5 лет после окончания высшего или среднего специального учебного заведения по специальности, полученной в этом учебном заведении.

Доплаты за кабинет и работающие компьютеры учитываются в среднем заработке во всех случаях, предусмотренных действующим законодательством (отпуск, пенсии и др.).

За ведение кружковой работы по техническому творчеству учащихся преподаватели могут оплачиваться по ставкам почасовой оплаты, предусмотренным для руководителей кружков, в пределах средств на заработную плату, имеющих в данном учебном заведении.

Аналогичным образом должны решаться вопросы, возникшие у преподавателей средних специальных учебных заведений.

59

Во втором номере журнала была опубликована статья В. Настасенко «Игровой пульт». Сегодня мы предлагаем вам программу для «Ямахи MSX-2» того же автора, иллюстрирующую способ использования джойстика.

```
10 *управление самолетом через джойстик
20 SCREEN1,2:COLOR2,1,1:KEYOFF:CLS:DJ=1
29 *формирование спрайтов
30 FOR K=0 TO 1:AA$=""
40   FOR I=1 TO 32
50     READ A: AA$=AA$+CHR$(A)
60     NEXT: SPRITE$(K)=AA$
70 NEXT
80 X0=127: Y0=160: *координаты самолета
90 ON STRIG GOSUB 170,170,170
91 STRIG(DJ) ON
100 PUTSPRITE 0,(X0,Y0).8
110 ON STICK(DJ) GOSUB 130,130,140,
      140,150,150,160,160
120 GOTO 110
130 Y0=Y0-1: GOTO 100
```

```
140 X0=X0+1: GOTO 100
150 Y0=Y0+1: GOTO 100
160 X0=X0-1: GOTO 100
170 FOR I=Y0 TO 0 STEP -1
180   PUTSPRITE 1,(X0,I).10
190 NEXT: PUTSPRITE 1,(0,209): RETURN
200 DATA 1,1,1,1,3,3,3,134,140,22,254
201 DATA 255,247,227,193,128,0,0,0
202 DATA 128,128,130,194,98,118,254,254
203 DATA 222,142,6,2
204 *самолет
210 DATA 0,0,0,0,0,0,0,0,0,0,128
211 DATA 128,128,128,0,0,0,0,0,0,0,0
212 DATA 0,0,1,1,1,1
213 *пули
```

ЧТО? МОЖЕТ ЭВМ

Цвет «в кармане»

За рубежом цветной жидкокристаллический дисплей становится все более обычной принадлежностью портативных компьютеров. Дисплей компьютера PC-9801LX5C имеет размер 23 см по диагонали, разрешающую способность 640×400 точек и может отображать 8 цветов; дисплей фирмы Sharp способен отображать уже 512 цветов в 640×480 точек; а более простой и дешевый дисплей компьютера PC-286LST фирмы Seiko-Enson обычно работает в черно-белом режиме (восемь градаций серого), но может работать и в псевдоцветном, используя светло-зеленый, красный и желтый цвета.

Новый век фотографии

На заре своего развития фотография считалась дорогим удовольствием. Виною этому была не только сложная опτικο-механическая аппаратура, но и фотоматериалы, при изготовлении которых даже сегодня невозможно обойтись без серебра.

Фотоаппарат фирмы «Канон», использующий принципиально новый способ фиксации изображений, не только обходится без привычной всем фотопленки, но и полностью исключает из процесса получения изображения химические операции по обработке фотобумаги, да и саму фотобумагу. Вместо всего этого в фотоаппарате используется 2-дюймовый гибкий магнитный диск, на который можно неоднократно записывать до 50 цветных видеокадров.

Фотоаппарат позволяет сразу же просматривать только что отснятый материал с помощью телемонитора, стереть неудачные кадры и записать на их место новые. А для того чтобы получить понравившееся изображение на бумаге, «отснятую» дискету нужно

вставить в цветной лазерный видеопринтер «Канон» и распечатать в нужном масштабе.

Большую помощь фотоаппарат с магнитным принципом записи изображений может оказать и деловым людям. Подобно ручному сканеру, он может сформировать на дискете изображение документа или чертежа, которое затем можно передать по сети в персональный компьютер или настольную издательскую систему для дальнейшей обработки и включения в отчет или книгу.

Что такое ЖКИ

Все чаще дисплеи компьютеров делают на основе ЖКИ. Расшифровать эту аббревиатуру (жидкокристаллический индикатор) несложно, но не все знают, что такое современный индикатор на жидких кристаллах.

Самый простой ЖКИ, знакомый по электронным наручным часам, состоит из двух стеклянных пластинок, зазор между которыми (очень малый, порядка 10 микрон) заполнен жидким кристаллом — особой жидкостью, продолговатые молекулы которой группируются с некоторой регулярностью. На внутренних поверхностях пластинок нанесен слой металла: на нижней пластинке сплошной и непрозрачный, а на верхней — тонкий до прозрачности, в виде отдельных электродов, имеющих форму тех знаков, отображать которые должен ЖКИ. При подаче напряжения на нижний электрод и один из верхних молекулы жидкого кристалла под действием электрического поля переориентируются и зазор между электродами становится непрозрачным. На индикаторе высвечивается знак.

Главные недостатки таких индикаторов — малые контрастность и угол обзора. Преодолеть их помогает твист-эффект.

Если стеклянную поверхность слегка потереть, на ней остаются микроцарапины. Молекулы жидкого кристалла стремятся ориентироваться параллельно им. Если царапины на одной стеклянной обложке ЖКИ перпендикулярны царапинам на другой, то между ними жидкий кристалл как бы скручивается: по мере передвижения от одной обложки к другой ориентация молекул плавно изменяется на 90° . Такая структура

поляризует проходящий свет, и использование поляризационного фильтра позволяет увеличить контрастность ЖКИ. Увеличивается и угол обзора.

Есть еще супертвист-эффект: он «закручивает» жидкий кристалл на 180° , а то и на 270° , значительно улучшает видимость изображения, и в итоге ЖКИ становится способен заменять телевизор: верхние электроды можно сделать очень маленькими (площадью $0,01 \text{ мм}^2$) — и все равно непрозрачность под ними будет ясно видна. Несколько тысяч электродов — и вот уже на индикаторе можно отображать произвольные фигуры.

Следующий шаг — цветное изображение. Его можно получить, нанеся на верхнее стекло вдобавок к множеству электродов множество цветных точек-светофильтров красного, голубого и зеленого цветов, сделав прозрачным нижний электрод и организовав подсветку экрана.

Огромным числом микроэлектродов управлять непросто. Эта проблема решается путем интеграции экрана и управляющей им электроники: нижний электрод изготавливают из кремния, прямо в нем формируют структуры, распределяющие электрические токи. И вот итог — современный цветной ЖКИ-индикатор: покрывное стекло с полярофильтром, под ним — прозрачные металлические электроды, ниже — цветные светофильтры, еще ниже — пленка ориентата, закручивающего молекулы жидкого кристалла, сам жидкий кристалл, снова пленка ориентата, слой кремния с электронными структурами и, наконец, нижняя стеклянная подложка и лампа подсветки.

Все очень просто.

У последнего предела

Американская компания Chips and Technologies продемонстрировала свою последнюю разработку — плату размером около $7 \times 60 \times 100$ мм, на которой размещены все компоненты IBM XT-совместимого компьютера: процессор 8086, ОЗУ на 512К байт, адаптер цветной видеографики, контроллеры гибкого и жесткого дисков, последовательные и параллельные интерфейсы.

Сейчас компания занимается разработкой аналогичного устройства на процессоре 80286.

Ю. ЗАЛЬЦМАН

Алма-Ата

Архитектура и программирование на языке ассемблера БК-0010

Микро-ЭВМ «Электроника БК-0010» серийно выпускается в СССР несколькими предприятиями с 1985 г. Не очень рискуя ошибиться, можно сказать, что парк БК-0010 сегодня по численности превосходит все остальные персональные ЭВМ, эксплуатирующиеся в СССР, вместе взятые.

Что же представляет собой наш самый массовый компьютер? О БК писали много и по-разному. Писали в специальной литературе и в газетах, хорошо и плохо, профессионалы и дилетанты. Среди лиц, избалованных мощной импортной техникой, распространено презрительное отношение к БК как к «большому калькулятору», «уродцу в семействе ЭВМ» и т. п. Но думается, что подобные отзывы исходят в основном от людей, либо совсем не знакомых с БК, либо пытавшихся с ним работать, но не имеющих навыков программирования и вследствие этого просто не справившихся с «калькулятором». Такие люди привыкли к тому, что ЭВМ все делает за них сама, а на дискетах имеется полный набор первоклассных фирменных программ — графические и текстовые редакторы, трансляторы языков высокого уровня, базы данных, электронные таблицы... Пользователю надо лишь освоить работу в операционной системе (ОС), и можно писать программы, например на Паскале. Или составлять каталог своей личной библиотеки. Или готовить статью в «Литературную газету» о том, какая это плохая ЭВМ — БК-0010 и какие плохие люди ее делают...

Но бог с ними. Мы все знаем, что хороший компьютер лучше, чем плохой. Однако на сегодняшний день наша страна, увы, сильно отстает от передовых

компьютерных держав мира. Так что все-таки БК-0010 такое? И так ли уж он плох?

Уважаемый читатель, нравятся ли вам микро-ЭВМ семейства ДВК: ДВК-1, ДВК-2, ДВК-3, наконец, ДВК-4? Если вы работали с ними, ответ скорее всего будет либо положительный, либо нейтральный — дело вкуса. Но никто, конечно, не скажет, что ДВК-2М, например, это большой калькулятор. Машина как машина... Ну, а теперь внимание: БК-0010 не только по системе команд полный аналог ДВК, но и по многим другим параметрам! БК-0010, как и ДВК, изготовлен на базе микропроцессорного комплекта К1801. Имеет такой же центральный процессор (К1801ВМ1), как и ДВК-1, такое же быстродействие (300 тыс. операций регистр-регистр в секунду), такое же адресное пространство (64К байта). Значит, можно сказать, что БК-0010 — это практически ДВК-1? Не совсем. БК-0010 лучше!

Да, его ОЗУ пользователя всего 16К байт. Но зато его экранное ОЗУ входит в общее адресное пространство и с экраном можно работать, как с ОЗУ пользователя. Вы не понимаете, что это за преимущество?! Довольно отметить, что это позволяет очень легко и просто организовать работу ЭВМ в графическом режиме. Кроме того, очень многие задачи (начиная с компьютерных игр и кончая системами распознавания образов) позволяют использовать экранное ОЗУ БК-0010 как часть ОЗУ пользователя и решать задачи прямо на экране, не занимая основную память ЭВМ. В БК-0010 реализована черно-белая графика довольно высокого разрешения — 512×256 точек, вдвое выше,

чем на такой популярной за рубежом ЭВМ, как «Синклер». Заметим, что ДВК-1 и ДВК-2 графического режима не имеют! БК-0010 имеет цветной режим, позволяя выводить на экран цветного монитора 4 основных цвета — красный, зеленый, синий и черный, причем программным путем можно легко увеличить число цветов до 8—16. БК-0010 не имеет дисковода. Это серьезный минус, но это и плюс! Отсюда его относительно низкая стоимость и высокая надежность. Монитор-драйверная система (МДС, или та же ОС) БК-0010 записана в постоянном запоминающем устройстве (ПЗУ) компьютера и не требует поэтому загрузки. Она не «зависает», практически не дает сбоев, нечувствительна к вирусам. В состав ОС (МДС) БК-0010 помимо основных программ, обеспечивающих работу ЭВМ, ее связь с внешними устройствами и т. д., входят два интерпретатора языков высокого уровня — Бейсик-MSX (по возможностям близкий к Basic-MSX микро-ЭВМ «Ямаха-2») и Фокал — очень простой и гибкий язык, предназначенный для решения научно-технических задач, к сожалению, до сих пор малоизвестный программистам. Фокал БК-0010 имеет расширенный набор средств, в том числе и графические операторы, и вполне может потягаться с Бейсиком по своим возможностям. И все эти средства доступны сразу после включения ЭВМ! Вам даже не нужно устанавливать в дисковод дискету и обрабатывать с помощью ОС. Есть в БК-0010 и так называемая мониторная система диагностики — МСД. Она не только дает возможность исчерпывающе проверить работоспособность и исправность ЭВМ, но и имеет средства, позволяющие писать, запускать и отлаживать программы непосредственно в машинных кодах. БК-0010 имеет очень большой набор шрифтов и псевдографических символов, может выводить на экран дисплея два формата текста — 64 и 32 символа в строке...

Но опытный пользователь персональных ЭВМ уже давно, наверное, с усмешкой читает этот панегирик БК-0010. У него готов вопрос: «А прикладное программное обеспечение? Ведь извест-

но, что за рубежом его стоимость обычно в 5—10 раз превышает стоимость «железа», т. е. самого компьютера! А чем может похвастать БК?» Может, уважаемый оппонент, и очень многим! На сегодня для БК-0010 разработано свыше 3000 прикладных, системных и инструментальных программ. На БК адаптированы (или, как говорят, «поставлены») языки: Бейсик (около 10 версий), Форт (3—4 основные версии), Си (2 основные версии), Т-язык. Ведется работа по адаптации языков Паскаль и Лисп. На БК имеется не менее 30 версий редакторов текста (на любой вкус); столько же, если не больше, графических редакторов; очень удобные и мощные ассемблер-системы. И конечно, игры. Всякие — логические (в том числе шахматы), динамические, игры-приключения, с графикой и текстовые, с реалистическими изображениями и звуковыми эффектами, черно-белые и цветные, полиэкранные, многосерийные и всякие другие, какие только можно вообразить. Ну а что касается математических программ (статистическая обработка данных, вычисления по формулам), то их такое множество, что автор не решается даже перечислить их основные классы. Многие из них представляют результаты обработки данных и вычислений не только в цифровом виде, но и графически, в том числе в цвете. А как с базами данных? Есть, и несколько версий. Есть и специальные программы сортировки по алфавиту, и драйверы для печати на принтерах разных систем, от «Консул-254» до «Роботронов» и «ThinkJet», и электронные таблицы. Есть программы упаковки и вывода изображений, программы уплотнения записей на магнитной ленте (МЛ) и даже синтезатор речи с неограниченным словарем русского языка!

Однако тут самое время остановиться, ведь даже простое перечисление имеющих для БК программных средств займет не одну страницу, а может быть, даже не один журнал! Пусть-ка лучше автор ответит на вопрос: откуда они взялись?

Что ж, вопрос законный. Почти все перечисленные средства написаны не

профессиональными программистами, а любителями. Но это в большинстве случаев не умаляет их достоинств. Ведь что такое профессиональный программист (если, конечно, это действительно программист, а не ремесленник)? Просто любитель, получающий зарплату за свое увлечение. Будем смотреть на вещи трезво — почти никто из профессиональных программистов не имеет специального программистского образования... И почти никто из имеющих такое образование программистом не работает! Эта парадоксальная ситуация, видимо, объясняется тем, что программирование — скорее призвание, талант, чем банальная профессия. Но если в театральное училище никогда не примут человека, лишённого артистического дарования, то в технический вуз или университет принимают, увы, всех, сдавших экзамены...

Итак, ассортимент разработанных для БК-0010 программ довольно широк и многие из них сделаны на высоком профессиональном уровне. Почему же до сих пор наблюдается острый дефицит программных средств? Дело в том, что в СССР нет законодательства, защищающего авторские права программиста, в связи с чем авторы не заинтересованы в распространении своих программ. Нет и единой системы распространения, единых расценок и гарантий. Многие государственные и кооперативные предприятия, торгующие программным обеспечением, делают это без всякого согласия авторов и не выплачивают им вознаграждения. Понятно, что авторы при этом даже препятствуют распространению программ, а те программы, которые тиражируются, зачастую представляют собой устаревшие версии или имеют ошибки, в том числе и намеренно внесенные автором. Исправить положение может только принятие соответствующих законов, а пока их нет — подчеркнутая добросовестность организаций, торгующих программами, по отношению к их авторам. Сегодня есть целый ряд организаций, тиражирующих программы по договорам с авторами и предлагающих пользователям за весьма умеренную плату лучшие образцы творчества программистов, пишущих

для БК-0010; пользователю нужно только следить за рекламой.

Теперь, после несколько затянувшегося, но, по-видимому, необходимого вступления, перейдем к вопросу; чего же все-таки не хватает БК-0010? Прежде всего, конечно, памяти. Хотя в настоящее время разработано уже несколько подходов для расширения ОЗУ и ПЗУ БК-0010, вряд ли можно ожидать, что их применение станет массовым, если промышленность не наладит выпуск таких «расширителей». Кроме малого объема ОЗУ пользователи БК-0010 также жалуются на низкое быстродействие. Но обе эти жалобы относятся в основном к программированию на языке высокого уровня. Действительно, Фокал — весьма медленный язык, а Бейсик оставляет желать лучшего в распределении памяти и оптимальности написания программ. Однако есть способ, позволяющий резко поднять скорость исполнения программ и (в большинстве случаев) существенно экономить память БК. Это переход к программированию в машинных кодах, или, что, по существу, одно и то же, на языке ассемблера.

Когда можно сказать, что пользователь БК-0010 «созрел» для этого шага? Если, во-первых, он ставит перед собой задачу, неразрешимую средствами Бейсика или Фокала. Если, во-вторых, он знает основы информатики и программирования и практически программирует хоть на одном из языков высокого уровня. В-третьих, как уже было сказано выше, нужны некоторые способности (но не большие, чем для программирования на Бейсике), а главное — большое желание и терпение, необходимые для всякого дела, а для работы на ассемблере — в особенности.

В данной публикации автор ставит перед собой задачу дать читателю возможность изучить архитектуру БК-0010 и познакомиться с основами программирования на самом гибком и мощном из языков любой ЭВМ — языке ассемблера.

По возможности будут поясняться также некоторые общие вопросы информатики и устройства ЭВМ, но объем журнального цикла ограничен, поэтому предполагается, что эти общие вопросы

читатель изучит самостоятельно с помощью имеющейся обширной литературы. Материал излагается как можно более доходчиво, что иногда влечет за собой некоторую нестрогость, расплывчатость формулировок и прочие дефекты.

Автор приносит искреннюю благодарность всем программистам и пользователям БК-0010, с которыми он вел переписку и лично встречался за последние четыре года, за их ценные советы, предложения, вопросы и пожелания, которые, собственно, и привели к мысли написать руководство по программированию на БК-0010 и без которых полное изложение материала, по-видимому, вообще было бы невозможно. Хотелось бы особо упомянуть программистов С. А. Кумандина и Д. Г. Гаева (Москва), А. В. Цаплева и А. Г. Makeева (Ленинград), В. А. Михайлова (Алма-Ата), С. И. Никанорова (Керчь).

Данное руководство может быть полезным не только программистам и пользователям БК-0010, но и пользователям других ЭВМ, имеющих аналогичную систему команд и похожий язык ассемблера («Электроника-60», ДВК, СМ-3, СМ-4, PDP-11 и т. п.)

Рекомендуемая литература

1. Руководства и пособия, входящие в комплект БК-0010 (различные предприятия, выпускающие БК, комплектуют его различными пособиями и под несколько отличающимися названиями, но примерно одинакового содержания).

2. *Вигдорчик Г. В., Воробьев А. Ю., Праченко В. Д.* Основы программирования на ассемблере для СМ ЭВМ. М.: Финансы и статистика, 1983.

3. *Сингер М.* Мини-ЭВМ PDP-11: программирование на языке ассемблера и организация машины. М.: Мир, 1984.

4. *Брусенцов Н. П.* Микрокомпьютеры. М.: Наука, 1985.

5. *Фрэнк Т. С.* PDP-11. Архитектура и программирование. М.: Радио и связь, 1986.

6. *Брябрин В. М.* Программное обеспечение персональных ЭВМ. М.: Наука, 1988.

7. *Осетинский Л. Г., Осетинский М. Г., Писаревский А. Н.* Фокал для микро- и мини-компьютеров. Л.-д: Машиностроение, 1988.

8. *Лин В.* PDP-11 и VAX-11: Архитектура ЭВМ и программирование на языке ассемблера. М.: Радио и связь, 1988.

Данным списком вовсе не исчерпывается возможная литература. Просто ее

перечень так велик, что привести все источники не представляется возможным. Ценные сведения нередко встречаются и в периодических изданиях — журналах «Информатика и образование», «Микропроцессорные средства и системы» и др.

Ваши отзывы о публикации можете присылать автору по адресу: 480091, Алма-Ата, ул. Джамбула, 55/57, кв. 8. Тел. 69-17-97.

Далее публикация делится на две части: архитектура БК и ассемблер БК. Они печатаются параллельно — именно так рекомендуется их читать.

Архитектура БК-0010

Что такое архитектура ЭВМ? По аналогии с архитектурой города можно догадаться, что это общее построение аппаратных средств машины. Действительно, город, состоящий из улиц и домов, построенных из кирпичей, имеет архитектуру, т. е. эти кирпичи расположены определенным образом. Жителей города обычно не интересует, что это за кирпичи, их химический состав, размеры и т. п. Зато очень важны расположение районов, ширина улиц, удобство и размеры домов и прочее, с чем мы постоянно имеем дело. Немаловажным вопросом является и городской транспорт! Чтобы узнать, из каких материалов построен дом, его надо разобрать, а это уже дело не его жителя, а строителей или ремонтников. Жителя же интересует обычно то, что он может увидеть, не прибегая к помощи отбойного молотка. Итак, архитектура ЭВМ — это ее аппаратные средства, доступные программным путем. Если программист не может обратиться к каким-либо средствам ЭВМ, то они к архитектуре не относятся.

Любая ЭВМ предназначена для обработки информации, причем, как правило, осуществляет эту обработку опосредованно — представляя информацию в виде чисел. Для работы с числами машина имеет специальную, важнейшую часть — центральный процессор (ЦП). Это универсальное логическое устройство, которое оперирует с двоичными числами, осуществляя простейшие логические и математические операции — сдвиг, сложение, сравнение и т. п., и не просто как придется, а в соответствии с программой, т. е. в заданной последовательности.

Чтобы оперировать с данными, их нужно откуда-то брать, значит, они должны где-то храниться. Также где-то должна храниться и необходимая последовательность действий ЦП — программа. Для хранения служит *запоминающее устройство* — ЗУ. ЗУ бывают *постоянные* (ПЗУ), в которых информация хранится, не изменяясь, сколь угодно долго, и *оперативные* (ОЗУ), информация в которых может быть в любой момент изменена в соответствии с результатами ее обработки.

Информация в ПЗУ записывается при их изготовлении, а для ввода информации в ОЗУ служат *внешние устройства* — клавиатура и накопитель на магнитной ленте (НМЛ). Другое внешнее устройство — дисплей служит для *вывода* информации. Вывод может осуществляться и на НМЛ, после чего информация там может храниться неограниченно долго и снова использоваться.

Как же ЦП общается с ОЗУ, ПЗУ и внешними устройствами? Через так называемое *адресное пространство* ЭВМ, в котором каждое из внешних устройств (а также каждая ячейка памяти) имеет свой адрес, подобно тому как имеет адрес каждый дом в городе. Но обычно, войдя в дом, мы еще не достигаем конечной цели — нужно найти квартиру, а возможно, и позвонить нужное число раз. Или представьте, что, попав в нужный дом, мы обнаружили, что там почтовое отделение, а в нем своя система адресов и свои правила работы. Есть такие «дома» и в ЭВМ — это *регистры внешних устройств*, или *системные регистры* (СР). Обратившись к такому адресу, мы как бы выходим за пределы нашего города, перед нами открыт весь мир! Но, чтобы общаться с ним, мы должны знать правила — как написать на конверте адрес, куда конверт опустить, а когда получим ответ — как его понять, и вообще, куда он придет и каким будет.

Такова в общих чертах архитектура БК-0010. Но мы рассмотрели ее очень поверхностно и схематично. Чтобы перейти к подробному рассмотрению, придется чуть отвлечься и разобратся: с какой же информацией и как ЭВМ работает?

Что такое системы счисления, вы, наверное, знаете. Привычная нам десятичная система очень неудобна для ЭВМ — ведь для обработки каждого числа нам нужно иметь 10 устройств, каждое из которых «знает» одну цифру из десяти — 0, 1, ..., 9. Однако любое число можно выразить и проще — в двоичной системе, где есть всего две цифры — 0 и 1. Правда, при этом запись числа становится очень длинной и неудобной для человека, а перевести двоичное число в десятичное весьма сложно. Есть компромисс —

восьмеричная система, где запись чисел достаточно компактна, а перевод в двоичную и обратно очень прост. В этой системе мы и будем общаться с ЭВМ, не забывая, однако, о том, что сама ЭВМ работает только с двоичными числами. Кратко напомним правила перевода двоичного числа в восьмеричное. Для этого нужно разбить двоичное число справа налево на тройки чисел (триады), а затем заменить каждую тройку на соответствующую восьмеричную цифру по таблице, которую легко запомнить.

| Двоичное число | Восьмеричная цифра |
|----------------|--------------------|
| 000 | 0 |
| 001 | 1 |
| 010 | 2 |
| 011 | 3 |
| 100 | 4 |
| 101 | 5 |
| 110 | 6 |
| 111 | 7 |

Таким образом, если дано двоичное число 1010100011111101, то разбиение на триады даст 001 010 100 011 111 101, а перевод в восьмеричную форму — 124375.

Мы дополнили двоичное число слева двумя незначащими нулями, чтобы получить полную триаду, но это необязательно. Перевод из восьмеричной системы в двоичную делается в обратном порядке и так же несложен.

Итак, все наши цифровые данные в дальнейшем мы будем приводить в восьмеричных числах, а в тех случаях, когда необходимо использовать десятичное число, будем помечать его буквой *d*. Заметим кстати, что восьмеричные числа от 0 до 7 равны десятичным и безразлично, в какой системе они приведены. Если же в составе числа встречаются цифры 8 и 9, то ясно, что число десятичное; в пометке *d* оно не нуждается. Если нужно будет привести двоичное число, то это будет специально оговорено. Другие системы счисления мы использовать не будем.

Теперь несколько слов о нашем «городском транспорте». Существуют две основных *конфигурации* ЭВМ — с отдельными шинами адреса и данных и с общей шиной. По названию фирм, развивающих то или иное направление, они носят название Intel и DEC соответственно. Не останавливаясь на преимуществах и недостатках каждой системы (а они есть и у той, и у другой), скажем лишь, что БК-0010 имеет конфигурацию DEC, или *общую шину адреса и данных*; другими словами, и адрес и данные передаются в ЭВМ по одним и тем же линиям,

связывающим ее блоки. Но ЭВМ построена так, что программист не ощущает особенностей ее организации и ему не так уж важно, как она работает; он может манипулировать адресом и данными отдельно и независимо, не путая их друг с другом. Одним словом, конфигурация ЭВМ (определяемая, кстати, в основном типом ЦП) к архитектуре ЭВМ (в нашем определении) не относится и поэтому мало нас интересует. Продолжая сравнение с городом, можно сказать, что транспорт — такси: достаточно сесть и назвать адрес, и нас по нему доставят. Однако в конце «поездки» придется платить, а «плата» на ЭВМ — это время. Если мы укажем адрес не оптимально («в центр — через пригород!»), «счетчик» может «нащелкать» слишком много. Косвенно мы все-таки учитываем конфигурацию ЭВМ через *способы адресации*, а как раз они-то в конфигурации DEC чрезвычайно гибки и разнообразны. Но об этом в свое время.

Продолжение следует

Ассемблер БК-0010

Вы, конечно, знакомы с имеющимися на БК-0010 так называемыми *языками высокого уровня*. Это Фокал и Бейсик MSX, которые находятся в ПЗУ (такое программное обеспечение (ПО), «зашитое» в ПЗУ, называется *резидентным*). В составе ПО БК-0010 имеются и другие языки программирования, но в отличие от резидентных они загружаются в ОЗУ. Все эти языки («вилльюрские» Бейсик-85 и Бейсик-87, Т-язык, Форт-83 и т. п.) имеют общую черту: это *языки-интерпретаторы*. Что это значит?

Программа на языке интерпретируемого типа пишется в виде текста, или *листинга*, в котором с помощью условных символов, цифр и слов указывается последовательность действия ЭВМ. Но хорошо известно, что единственный язык, непосредственно доступный ЭВМ — язык машинных команд, двоично-восьмеричный код. Написанная на языке высокого уровня программа не может быть исполнена непосредственно, а лишь с помощью программы-интерпретатора (входящей в состав языка), причем при исполнении и листинг переводится интерпретатором в последовательность машинных команд, которую и исполняет ЭВМ. Программа хранится в памяти ЭВМ практически в том виде, в котором написана (небольшая модификация листинга принципиального значения не имеет). Если нужно еще раз исполнить программу, интерпретатор снова переводит ее в машинные команды, причем не всю сразу,

а по частям (обычно по одному оператору). Преимущества интерпретатора — простота языка и удобство программирования; программу можно просмотреть в той же форме, в которой она написана, исправить, дополнить и т. п.; компактность представления программ в ОЗУ в виде листинга. Недостатки — низкая скорость исполнения программ (перевод в машинные коды каждый раз выполняется заново) и то, что программа-интерпретатор всегда должна присутствовать в памяти.

Помимо языков интерпретируемого типа существуют так называемые *транслируемые языки*, например Фортран, Паскаль, Кобол, Си, ПЛ/1, Ада. Программа на таком языке пишется также в виде листинга, а затем *транслируется*, т. е. переводится в машинные команды, но при этом не исполняется. При таком переводе вместо одного оператора, например на Фортране, в программу могут быть вписаны (как говорят, *сгенерированы*) десятки и даже сотни машинных команд. В результате трансляции образуется программа в машинных кодах, так называемый *загрузочный модуль*. Поскольку это уже не листинг, а готовый машинный (или, как его еще называют, *объектный*) код, программа исполняется несравненно быстрее, чем на языке-интерпретаторе, а дальнейшее присутствие транслятора в памяти становится ненужным. К недостаткам языков этого типа относятся прежде всего то, что программа-транслятор, входящая в состав языка, при переводе листинга на язык машинных команд строит загрузочный модуль не всегда оптимально. В зависимости от конкретных транслятора, языка и программы эта неоптимальность может приводить к увеличению длины программ в 1,5—3 раза и соответствующему снижению возможного быстродействия по сравнению с программой, составленной в машинных кодах опытным программистом. Происходит это потому, что язык высокого уровня по структуре весьма сложен и при его трансляции всегда присутствует некоторая неоднозначность, когда можно сделать перевод и так, и эдак. По этой причине и обратный перевод с машинных кодов на язык высокого уровня чрезвычайно труден. Очень сильно влияет и то, что программы на языках высокого уровня должны быть *переносимы на уровне листингов*, т. е. должны исполняться по возможности одинаково на машинах разных систем. Получается, под один язык должны «приспособляться» разные машины, которым, как правило, свойственны жесткие архитектура и система команд, не всегда соответствующие средствам языка. Как разрешить это противоречие и как, с одной стороны, использовать все

ресурсы ЭВМ, а с другой — сделать программу оптимальной, а средства ее написания удобными для программиста?

На заре программирования программы писали непосредственно в виде последовательности машинных команд (в машинном коде), при этом программы, естественно, полностью соответствовали архитектуре ЭВМ, были оптимальны как по длине, так и по быстрдействию (конечно, при известной квалификации программиста). Но программирование в машинных кодах — каторжный труд, превращающий человека в машину; 90 % рабочего времени уходило не на собственно программирование, а на запоминание, вычисление и прочие операции, связанные с кодами. Вскоре догадались передать эти функции ЭВМ, освободившись тем самым от рутинного труда, да и мощность машин стала достаточной, чтобы помочь программисту. Тогда-то из машинного языка и родился язык *ассемблера*. Как и язык высокого уровня, он позволяет пользоваться системой обозначений, более удобной и легко запоминающейся по сравнению с машинными кодами; но в отличие от других языков это язык *низкого уровня*, или *машинно-ориентированный язык*. Язык ассемблера всегда приспособлен для машин определенного типа, рассчитан на систему команд конкретного процессора. Поэтому он позволяет использовать все без исключения ресурсы машины, достигая максимально возможных компактности и быстрдействия программ. Каждой команде на языке ассемблера соответствует команда в машинных кодах, поэтому нетрудно как прямой (с ассемблера в коды), так и обратный перевод. Структура языка проста, и программа-транслятор получается достаточно компактной. Все эти преимущества обусловили чрезвычайно широкое распространение языков ассемблера (на каждой ЭВМ он свой) и их высокую популярность. Тем более велика роль ассемблера на такой ЭВМ, как БК-0010, с ее далеко не безграничными возможностями как по быстрдействию, так и, особенно, по ресурсам памяти.

Итак, что же такое ассемблер? Во-первых, это машинно-ориентированный язык, система условных обозначений для программирования в машинных кодах. Во-вторых, ассемблером называют и специальную программу, обеспечивающую работу на языке ассемблера, только при этом обычно добавляют: «ассемблер-система».

А что же машинные коды, их знать уже не требуется? В общем, да. Ассемблер-система обеспечит перевод листинга программы в машинный код, а другая программа, называемая *дисассемблером*, может легко

выполнить обратный перевод машинных кодов на язык ассемблера, причем обе программы делают это быстро и безошибочно. Конечно, бывают случаи, когда знание машинного кода может оказаться весьма полезным, избавляя от необходимости применять специальные программные средства и даже экономя время. Но это случается относительно редко, поэтому изучение машинных команд не входит в нашу задачу, а желающие могут ознакомиться с машинными кодами по имеющейся обширной литературе. В частности, список машинных команд и в кодах, и на языке ассемблера (так называемый *мнемокод*) приведен в «Руководстве системного программиста», прилагаемом к БК-0010. Мы же по ходу изложения будем обращаться к машинным кодам только в редких случаях, по возможности ограничиваясь языком ассемблера.

67

Ассемблер-системы

Для БК-0010 создано несколько ассемблер-систем. Не останавливаясь на первых, уже устаревших программах, отметим, что все современные ассемблер-системы имеют ряд общих черт. Они состоят из четырех основных частей, объединенных в одном модуле: *монитора, редактора, транслятора и компоновщика*.

Монитор включается при запуске ассемблер-системы и обеспечивает с помощью ряда директив диалог с пользователем, позволяя считывать тексты программ с МЛ, записывать их на МЛ, транслировать и компоновать программы, а также (в наиболее совершенных системах) запускать готовый загрузочный модуль.

По особой директиве можно перейти в **редактор текста** ассемблер-системы. От совершенства этой части в большой степени зависит удобство работы в ассемблере. Редактор позволяет писать и редактировать тексты (листинги) программ. Мощный редактор ускоряет и облегчает работу в ассемблере многократно.

Транслятор запускается директивами монитора. Он транслирует (переводит в машинные команды) текст программы, подготовленный в редакторе или загруженный с МЛ, резервирует ячейки памяти под переменные, константы и массивы, вычисляет адреса строк программы и адреса переходов, а также выявляет ошибки, допущенные программистом при написании листинга.

Компоновщик, также работающий под управлением монитора, во-первых, позволяет таким образом обработать код оттранслированной уже программой, что она становится

приспособленной для работы в заданной конкретной области ОЗУ, а во-вторых, может сосуществовать в единое целое несколько модулей, загруженных с магнитной ленты, для их совместной работы.

Первыми ассемблер-системами, имеющими все вышеперечисленные признаки, были МИКРО.С (авторы — А. М. Сомов и С. В. Шмытов, Москва). Начиная с МИКРО.3С и кончая МИКРО.1103 (1989 г.), эти ассемблер-системы непрерывно совершенствовались, расширялась система команд, улучшался сервис. Однако основной их недостаток — довольно слабый редактор текста — сохранился до самой последней версии.

Заметный шаг вперед в области создания ассемблер-систем для БК-0010 был сделан, когда к работе подключился С. А. Кумандин (Коломна). Начав с усовершенствования существовавшей на тот момент версии (МИКРО.8С) и снабдив ее значительно более мощным редактором, что привело к появлению версии МИКРО.8К, он продолжал работу над ассемблер-системой, все дальше уходя от исходной версии. В результате в 1988—1989 гг. были созданы версии МИКРО.10К и МИКРО.11К, которые можно считать уже вполне самостоятельными ассемблер-системами, так далеко они ушли от МИКРО.8С. Если версия МИКРО.8К имела ряд ошибок, то МИКРО.10К является высококачественной, тщательно отлаженной программой, имеет расширенную систему команд, удобные дополнительные форматы записи псевдокоманд, а главное — редактор текста, который не уступает знаменитому редактору EDASP, а во многом и превосходит его. Не останавливаясь подробно на его достоинствах, можно только сказать, что многие считают его на сегодня вообще лучшим экраным редактором текста, созданным когда-либо для БК-0010, включая и специальные редакторы текста.

На базе МИКРО.10К создано еще несколько версий — МИКРО.11К, рассчитанный на «зашивку» в ПЗУ и имеющий еще более совершенный редактор; МИКРО.10—01К, рассчитанный на работу с клавиатурой БК-0010-01; МИКРО.10К-РП, редактор которого может работать как с обычным экраном, так и в режиме РП, что позволяет при необходимости иметь дело с текстами длиной до 50000. Следует отметить, что все версии МИКРО.10К позволяют в режиме РП компоновать в памяти тексты программ такой же длины, а это в большинстве случаев позволяет обойтись без промежуточных записей на МЛ объектных модулей при компоновке программ и значительно повышает оперативность и удобство работы.

В дальнейшем мы будем при описании ассемблера ориентироваться в основном на версию МИКРО.10К (V.251288), но при необходимости будем отмечать и особенности других версий. Подробно описывать какую-либо (а тем более все) версии ассемблер-систем, созданных для БК-0010, не представляется возможным. Приобретая конкретную версию, пользователь обычно получает и ее краткое описание с перечислением команд ассемблера, директив редактора и монитора и может освоить ее самостоятельно. Мы же будем рассматривать общие моменты программирования на ассемблере, действительные практически для всех версий. Но, еще раз повторим, лучшей версией сегодня, на наш взгляд, является МИКРО.10К. Некоторые из имеющихся в ее составе команд, директив и форматов записи отсутствуют в других версиях ассемблеров, однако это не должно огорчать не имеющих МИКРО.10К, так как практически все эти команды и форматы имеют пусть менее удобные, но вполне заменяющие их эквиваленты во всех версиях, начиная с МИКРО.8С.

Продолжение следует

И. ПАНЧЕНКОВ

Приручение БК

Наконец-то вы приобрели БК-0010, но... радоваться еще рано. У обнвыи есть множество мелких недостатков, и нужно изрядно попотеть, чтобы оказаться владельцем действительно надежного (почти безотказного) и удобного бытового компьютера. Для достижения этой цели не обязательно совершать длинный путь проб и ошибок, которым прошли первые владельцы БК; вы можете вос-

пользоваться их коллективным опытом, о котором я постараюсь рассказать.

Начнем с того, что в прямом смысле слова лежит на поверхности: с клавиатуры. Она бывает двух типов: пленочная, или низкая, и полноходовая, или высокая.

Пленочную клавиатуру нужно (поверьте опытному пользователю!) сразу разобрать, пленку с нарисованными клавишами и тонкую

резиновую прокладку с прямоугольными вырезами выбросить, а все полиэтиленовые толкатели клавишных кнопок покрыть каким-либо лаком и тщательно высушить. Затем можно (если есть желание) покрасить толкатели разноцветными нитроэмальями (например, в соответствии с цветами снятой пленки) и снова высушить. Наконец, нужно перевести на толкатели деколи (переводные изображения) клавиш БК, которые изготавливаются Химкинским многоотраслевым комбинатом (141400, Московская обл., г. Химки, Юбилейный просп., 40; тел. 571-12-75; возможна продажа по почте). Деколи производятся в белом, черном, красном, желтом, зеленом и синем цветах; учтите это при раскраске клавиш. Переведя деколи, еще раз покройте толкатели лаком — на этот раз масляным, чтобы не испортить изображения на клавишах, — и высушите в течение суток.

Теперь можно собрать клавиатуру «позаводскому», а можно внести небольшие, но очень полезные изменения.

Клавиша НР в заводском варианте расположена в левом дальнем углу, что делает невозможным одновременное нажатие одной рукой ее и какой-либо из клавиш правой части клавиатуры. Сначала это не мешает, но по мере увеличения навыка работы начинает раздражать. Поэтому рекомендую разорвать запараллеливающие монтажные соединения на печатной плате клавиатуры у входных кнопок в парах, обеспечивающих переключение регистров РУС и ЛАТ, и присоединить высвободившиеся кнопки параллельно кнопке НР (и друг другу). Нижний ряд клавиш приобретет вид, показанный на фото, и очень скоро вы поймете, что обмен части клавиш РУС и ЛАТ на НР весьма выгоден! При такой организации клавиатуры на ней не остается мертвых зон и она становится управляемой пальцами одной руки. Кстати, для такой переделки полезно запастись лишней парой деколей для «размножившейся» клавиши НР.

Теперь можно собрать клавиатуру, объединить какими-либо накладками четыре про-

Пленочная клавиатура после переделки



бельные кнопки и две кнопки ВВОДа и посмотреть, что же получилось. А получилась качественно новая клавиатура, лишенная ряда серьезных недостатков: взаимной чувствительности клавиш из-за общей покровной пленки, тактильной «слепоты» из-за ее амортизирующего действия и, наконец, нелепой мертвой зоны. Мелочи? Только до тех пор, пока нет нужды работать на компьютере по-настоящему.

Полноходовая клавиатура также требует доработки. Прежде всего следует отрегулировать ход клавиш. Для этого снимите пластмассовые колпачки с толкателей (они просто сдергиваются вверх). Внутри клавиш вы увидите пару контактов, расположенных по разные стороны толкателя, подвижные части которых имеют, в свою очередь, по паре «усиков». Заводская регулировка контактов такова, что при нажатии на толкатель «усики» скользят вниз по неподвижному контакту, проскальзывая по нему, вибрируя и совершая множественные касания, что и обеспечивает так называемый дребезг клавиш, т. е. многократное печатание символа при однократном нажатии на клавишу. Поэтому придется проделать нудную и кропотливую, но необходимую работу: подогнуть «усики» вниз и внутрь корпусов клавиш. Цель — во-первых, установить «усики» так, чтобы при нажатии на толкатель они вонзались в неподвижные контакты под прямым углом, как штык; во-вторых, сделать минимальным зазор между контактами. Это обеспечит минимальный холостой ход клавиш, и любое нажатие на толкатель, превышающее необходимое, будет только усиливать прижим подвижного контакта и препятствовать его произвольным, неконтролируемым прыжкам. Вторую пару «усиков», чтобы не возиться с регулировкой, лучше заизолировать от неподвижных контактов тонкой прокладкой из изолирующего материала.

Следующая операция — смазка всех трущихся частей внутри корпуса клавиши. Нелепая операция, но увы — они сделаны из плохо подобранных пластмасс! Для смазки рекомендую часовое масло с добавкой графитового порошка (например, из грифелей мягких простых карандашей). Будьте предельно аккуратны, не замаслите контакты!

Регулировка клавиатуры и уход за нею чрезвычайно важны, так как серьезная работа за плохой клавиатурой приводит к тяжелейшему заболеванию — тендовагиниту (воспаление кистевых сухожильных мешков). Он знаком мне не понаслышке, и потому настоятельно рекомендую: не пренебрегайте ничем, что может облегчить работу на клавиатуре!

Пора надевать на толкатели колпачки?

Еще нет. Во-первых, на них нанесены далеко не все необходимые обозначения. Во-вторых, на двух из них — 0 и Ъ — обозначения ошибочны. В-третьих, все обозначения сделаны неустойчивой к истиранию краской, и клавиатура постепенно «слепнет». Так что владельцам полноходовых клавиатур тоже придется приобретать деколи. Однако нанести дополнительные обозначения придется в основном на лицевых вертикальных гранях колпачков, так как клавиш в полноходовом варианте меньше, чем в пленочном. Обзор будет не слишком хорош, но это лучше, чем ничего... С клавиш 0 и Ъ нужно убрать символы фигурных скобок: их место на клавишах Щ и Ш. На клавиши ключей удобно нанести закодированную в них информацию, разместив ее справа по вертикали. Одновременно следует укрупнить обозначения знаков препинания, это делается с помощью тонкого сверла, острого ножа и черного нитролака.

70

Когда все обозначения нанесены, покройте колпачки масляным лаком, высушите, тщательно оберегая от пыли, и поставьте на место.

Полноходовая клавиатура имеет целых две «мертвых» зоны: по клавишам «широкая стрелка вниз» и AP2. Устранить их можно, встроив два микропереключателя типа MT-1 внутрь пробельной клавиши; кнопки переключателей должны выходить на лицевую вертикальную грань клавиши. Разомкнутые контакты переключателей подключаются тонким гибким проводом параллельно соответствующим клавишам, а в обрамлении клавиатуры (в корпусе компьютера) делаются вырезы в которые кнопки уходят при нажатии пробельной клавиши. Эта доработка полностью оправдала себя на практике.

Работа с клавишами окончена, но «лежащее на поверхности» не исчерпано. Вокруг клавиатуры рекомендую разместить таблицу-подсказку с мнемоникой комбинационных операций и таблицу кодов ASCII. В каждой

из трех частей последней способ нахождения кода одинаков: в качестве первой цифры кода берется номер части таблицы (0, 1 или 2), в качестве второй — номер столбца, в котором стоит нужный символ, в качестве третьей — номер строки. Например, код буквы J — 074, M — 237.

Последнее, что находится на поверхности, — светодиод, сигнализирующий о включении питания. По его поводу хочу посоветовать сам компьютер и всю его периферию (магнитофон, телевизор и пр.) с выключате-

| | | | |
|----------|---------|-------------|---------|
| AP2 ↓ | | СУ | |
| | | М | уст.таб |
| | | П | сбр.таб |
| 1 | красный | Р | ↩ |
| 2 | зеленый | Т | ↓ Г |
| 3 | синий | У | * ↩ |
| 4 | черный | Э | ↙ |
| | | Щ | ↗ |
| | | Ч | ↘ |
| | | Ь | ↙ |
| | | ВС | ↑ |
| AP2 | | | |
| | | ← | РЕД |
| | | : или ↑ | КУРСОР |
| | | ; или ↓ | 32/64 |
| | | , | ИНВ.С |
| | | — | ИНВ.Э |
| | | . | УСТ.ИНД |
| | | / | ПОДЧ. |
| | | СБР | РП |
| | | СУУ или СУЕ | ГРАФ |
| | | ↩/СУФ/СУЖ | ЗАП |
| | | ⇒/СУГ/СУВ | СТИР |

| | | | | | | | | | | | | | | |
|---|-------|--------|----------|----|---|---|---|---|---|---|---|---|---|-------|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 0 | 1 | 2 |
| 0 | ШГ | | ↓ ГТ | ↘ | С | 2 | < | F | P | Z | 0 | d | n | x |
| 1 | ПОВТ. | | ↩* | ↙ |) | 3 | = | G | Q | [| 1 | e | o | y |
| 2 | ИСУ | СБР | ← пробел | * | 4 | > | H | R | \ | 2 | f | p | z | |
| 3 | КТ | ВВОД | → | ! | + | 5 | ? | I | S |] | 3 | g | q | { |
| 4 | БЛР | РУС | ←+ | .. | , | 6 | @ | J | T | ↖ | 4 | h | r | |
| 5 | ГРАФ | ЛАТ | → | # | — | 7 | A | K | U | _ | 5 | i | s | } |
| 6 | ЗАП | СБРТАБ | ↑ | ⌘ | . | 8 | B | L | V | ` | 6 | j | t | — |
| 7 | ЗВУК | | ↓ | % | / | 9 | C | M | W | a | 7 | k | u | ■ |
| 8 | ← | ↩ | ↙ | & | ∅ | : | D | N | X | b | 8 | l | v | |
| 9 | ТАБ | ↑ ВС | ↘ | / | 1 | ; | E | O | Y | c | 9 | m | w | повт. |

лями в положении «включено» присоединить к достаточно мощному стабилизатору, а уже последний включать и выключать. Такой способ в максимальной степени защищает аппаратуру от бросков сетевого напряжения, гарантирует воспроизводимость режима работы, позволяет удалить блок питания БК на достаточно большое расстояние, чтобы ликвидировать его паразитное влияние на телевизор и коммутации компьютера с периферией.

А теперь начнем погружаться вглубь — займемся корпусом, блоком МСТД и процессорной платой. В корпусе самое интересное — несоответствие отверстий для штеккерных вилок диаметру последних. Из-за этого вилка не досылается в гнездо до упора и со временем разбалтывает его. Диаметры всех вилочных отверстий нужно увеличить до 18 мм.

Сзади слева в узкой щели разъема торчит внушительных размеров коробка блока МСТД. Внутри нее простенькая печатная

плата с двумя ПЗУ, двумя резисторами, тремя конденсаторами и ответной частью разъема — и все это весит чуть не 100 г и «съедает» 120 квадратных сантиметров стола, а со временем выводит из строя не такой уж крепкий разъем МПИ. А между тем блок МСТД нетрудно убрать в корпус БК и переходить с языка на язык простым нажатием двух кнопок.

Для этого разберем блок и аккуратно отпаяем вилочную часть разъема: и она, и коробка с винтами нам еще пригодятся. В плате с ПЗУ просверлим два крепежных отверстия диаметром 3,2 мм с расстоянием между центрами 88 мм; теперь ее можно укрепить на процессорной плате поверх разъема МПИ, но прежде разрежем дорожку, идущую от выводов ПЗУ за номером 23 там, где она подходит к общей шине, а на процессорной плате прорежем прямоугольное окно 2×7 мм прямо у ножек разъема МПИ линии В в том месте, где этому не мешает печатный монтаж обеих сторон платы: сюда пройдут провода, соединяющие разъем МПИ с тем, что осталось от блока МСТД. Удалим также с

Высокая клавиатура после переделки



| | | | | | | | | | | | | | |
|-----|--------|---------|---|---|-----|---|---|---|---|---|---|---|---|
| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2 | 0 | 1 | 2 | 3 | 4 | 5 |
| ИСУ | РП | ЗАП | Щ | Г | ••• | → | 0 | х | р | э | Ф | П | З |
| | | СТИР | ⊥ | Т | — | ■ | 1 | и | с | щ | Г | Я | Ш |
| БЛР | | РЕД | ♥ | ⊥ | ≡ | ю | 2 | й | т | ч | Х | Р | Э |
| | | СБР→ | ∟ | ↓ | ↑ | а | 3 | к | у | ь | И | С | Ш |
| | ШГ | курсор | ≡ | + | ◆ | б | 4 | л | ж | Ю | Й | Т | Ч |
| | красн. | 32/64 | └ | | └ | ц | 5 | м | в | А | К | У | Ь |
| | зелен. | инв.с | └ | └ | ≡ | д | 6 | н | ь | Б | Л | Ж | |
| | синий | инв.э | ≡ | ← | ≡ | е | 7 | о | ы | Ц | М | В | |
| | черный | уст.инд | ≡ | ≡ | ≡ | ф | 8 | п | з | Д | Н | Ь | |
| | граф | подч. | ♠ | ↑ | ≡ | г | 9 | я | ш | Е | О | Ы | |



Процессорная плата с блоком МСТД и переключателем языков

Блок МСТД отодвинут в сторону

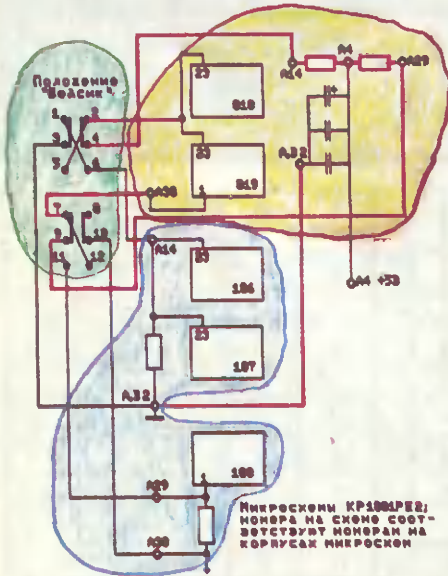


72

процессорной платы ползунковый переключатель (если он есть), перезапускающий процессор.

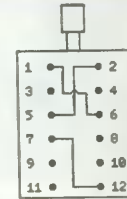
Теперь посмотрим на принципиальную схему (рис. 1) будущей переделки. Красным на схеме обозначены элементы МСТД, синим — процессорной платы БК; в зеленом

1



Точки монтажных соединений БК-0010 и МСТД

| разъема ХТЗ БК | Номера контактов | |
|----------------|------------------------------|---------------|
| | МСТД | переключателя |
| A4 | A4 | |
| A2, B2 | A2, B2 | 3 |
| A14 | | 6 |
| A23 | A23 | |
| A25 | A25 | |
| A26 | A26 | |
| A27 | A27 | |
| A28 | A28 | |
| A29 | | 11 |
| A30 | | 10 |
| A31 | A31 | |
| | A14 | 4 |
| | A29 | 9 |
| | A30 | 7 |
| | выводы 23 микросхем 018, 019 | 2 |
| B7 | B7 | |
| B20 | B20 | |
| B22 | B22 | |
| B23 | B23 | |
| B24 | B24 | |
| B25 | B25 | |
| B26 | B26 | |
| B27 | B27 | |
| B28 | B28 | |
| B29 | B29 | |
| B30 | B30 | |
| B31 | B31 | |
| B32 | B32 | |



2

прямоугольнике — новый элемент, переключатель языков. Облегчит переделку таблицы соединений. Рис. 2 показывает нумерацию выводов кнопочного переключателя языков (типа ПКнб1), который нужно объединить в единый блок с другим таким же, но лишенным стопорного механизма, служащим для перезапуска процессора после перехода на другой язык программирования.

Блок переключателей удобно расположить неподалеку от реле РС-15, управляющего работой магнитофона, между теми точками на плате, где ранее был расположен штатный переключатель, перезапускавший процессор. Кнопки выводятся через левую стенку корпуса и оказываются удобно защищенными его нависающей частью от случайного нажатия.

Теперь БК стал компактнее, а для перехода с языка на язык достаточно последо-

вательно нажать обе кнопки.

Прежде чем собрать компьютер, не полнитесь еще раз пропаять в нем все соединения (увы, качество заводской конвейерной пайки оставляет желать лучшего); жало паяльника должно быть при этом заземлено, его температура не выше 270 °С, в качестве флюса используйте раствор канифоли в спирте. Впоследствии ваши труды окупятся стoriцей.

Еще можно увеличить вдвое тактовую частоту процессора (эта переделка описана в «ИНФО» № 2). При этом можно воспользоваться точно таким же блоком кнопочных переключателей, у одного из которых нет стопорного механизма (SA2). Последовательность их использования такова: сначала переключателем SA2 приостанавливается работа процессора, для чего он переводится в положение, соответствующее нижнему на схеме, затем переключателем SA1 изменяется тактовая частота, и, наконец, возвращением переключателя SA2 в исходное положение процессор снова запускается. Помните, что переключение тактовой частоты при работающем процессоре приведет к сбою и потере информации и что с магнитофоном БК может работать только при нормальной тактовой частоте.

К сожалению, неизвестно, насколько данная переделка безопасна. Пока такие опыты кончались удачно...

Кстати, SA2 можно использовать в Бейсике для приостановки работы программы аналогично клавише ШАГ в Фокале.

Можно также повысить надежность срабатывания клавиши СТОП и обеспечить дополнительную защиту контактов клавиатуры от дребезга. Для этого достаточно увеличить емкость конденсатора С2 (см. схему БК) до 0,15 мкФ, конденсаторов С3 и С4 до 1,0 мкФ и сопротивлений R3 и R4 до 470 кОм.

Поставив последнюю точку, я загрустил: несмотря на простоту и полезность многих из описанных переделок, широко распространения, они, видимо, не найдут: ведь побочным их результатом будет отказ мастерских от ремонта вашего компьютера в случае его выхода из строя (весьма маловероятного, впрочем). Но, с другой стороны, накопленный опыт действительно ценен, и возможность пользоваться им нужно дать всем, кто к этому готов.

Итак, продолжим. Теперь настала очередь периферийных устройств. Если в качестве дисплея вы используете черно-белый телевизор, рекомендую закрыть его экран зеленым прозрачным фильтром (они продаются в магазинах бывшего ВТО, ныне Союза театральных деятелей, под названием прожекторных светофильтров). Зеленое свечение экрана

благоприятно для глаз — это доказано врачебными исследованиями и практикой. При хорошо подобранных контрастности и яркости изображения время непрерывной работы можно довести до 1,5—2 ч, но все же лучше делать 15—20-минутные перерывы после каждого часа работы; находиться от экрана надо на расстоянии вытянутой руки. Правила просты, но они позволят вам сохранить зрение и ясную голову во время работы — да и после нее.

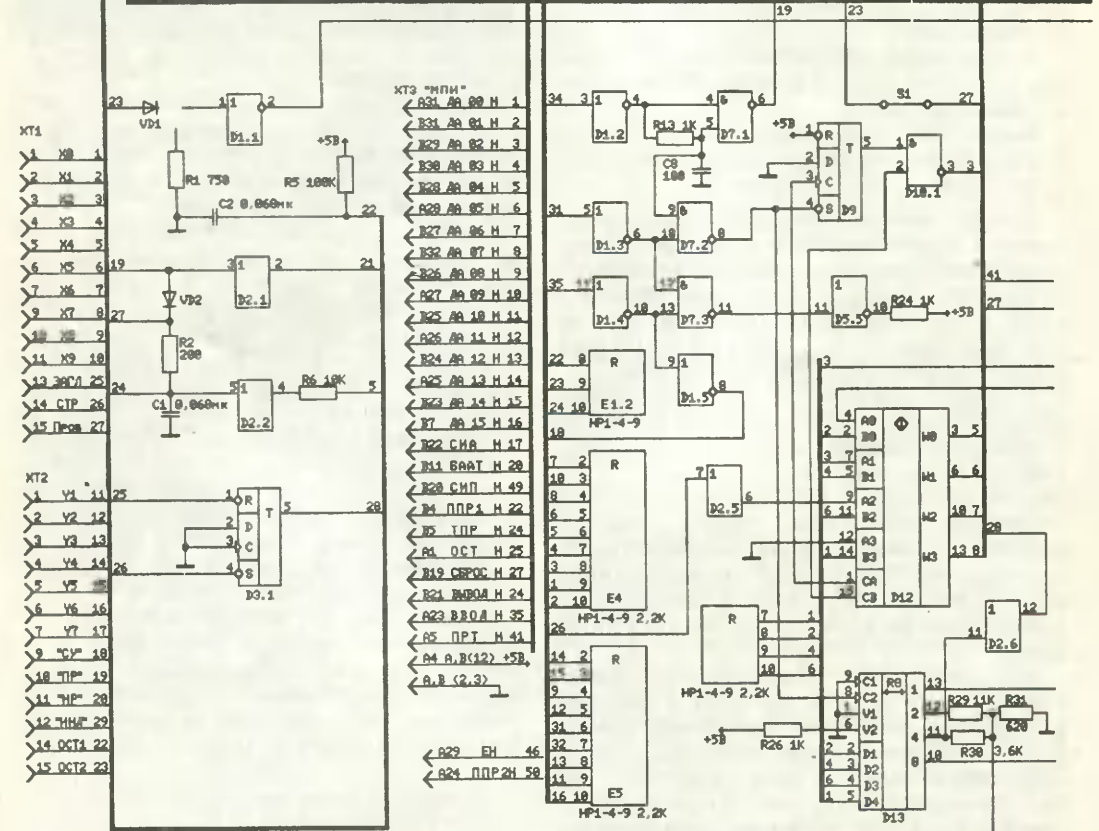
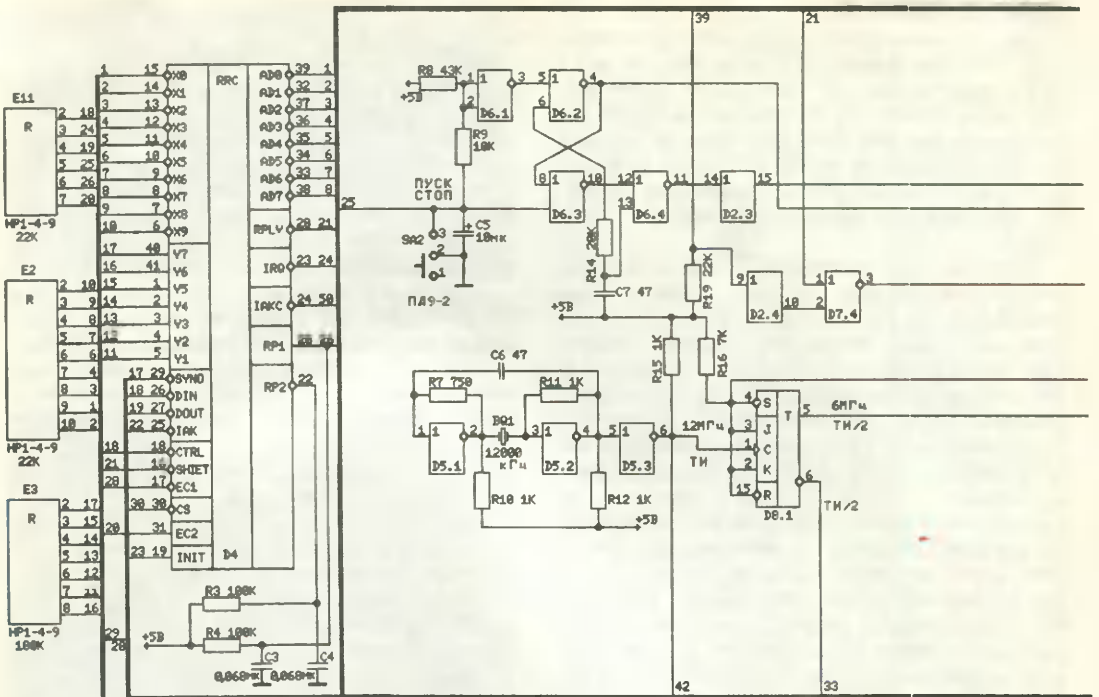
Я не стану рассматривать здесь способы подключения БК к телевизору — по этому поводу уже есть немало публикаций. Отмечу только, что применительно к наиболее распространенному телевизору «Электроника-404» оптимальное решение найдено П. Гужа (ИНФО. 1989. № 6); оно без труда может быть перенесено на любые марки портативных телевизоров.

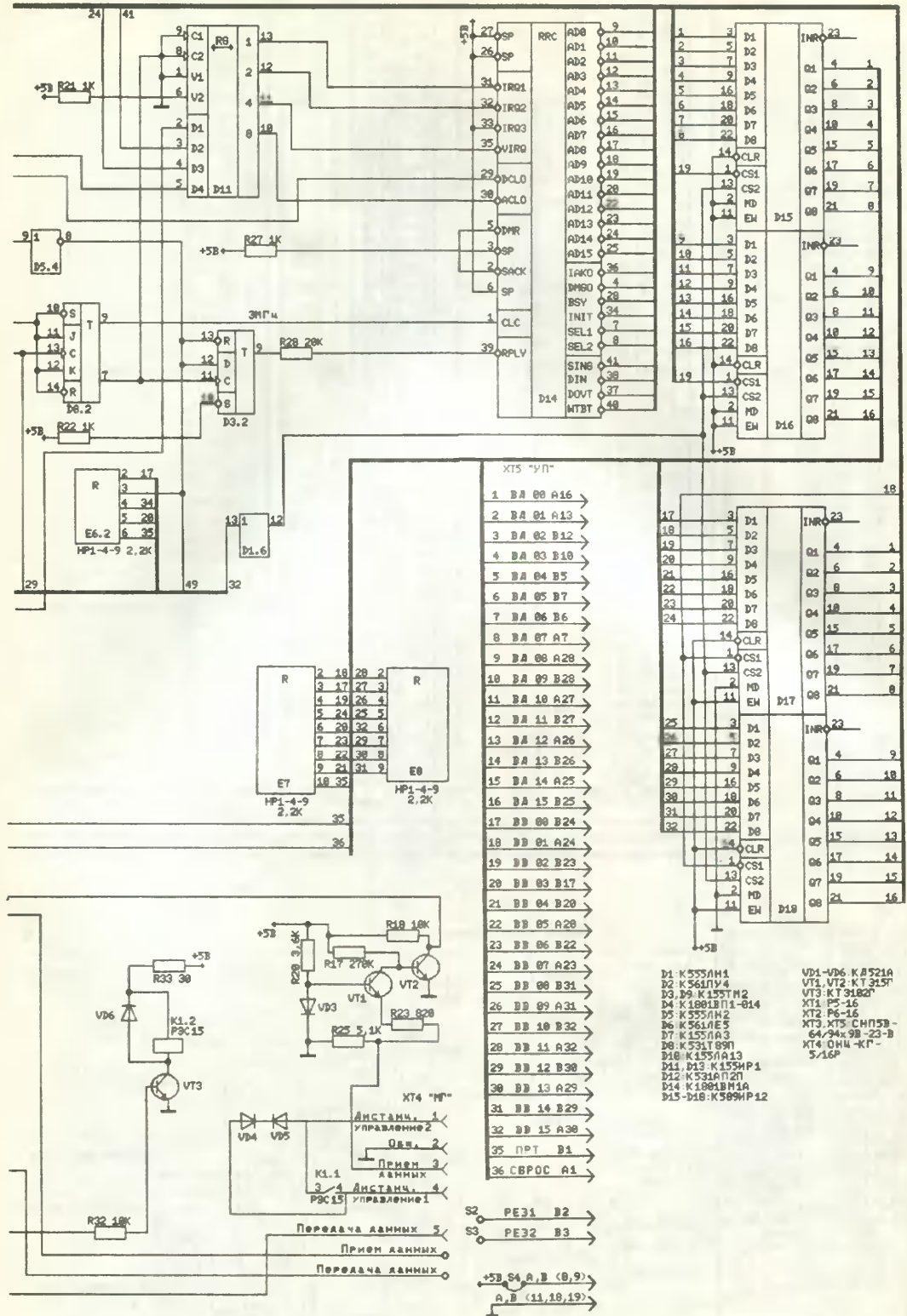
В наихудшем положении находятся те, кто использует в качестве дисплея цветной телевизор: хорошего изображения на его экране получить невозможно и использовать для работы с текстовыми файлами нельзя. Посоветовать можно только одно: при работе с текстами отключайте — программным или физическим образом — все электронные пушки кинескопа, кроме синей, а полным их набором пользуйтесь лишь при работе игровых программ.

Самая общая и важная рекомендация в отношении работы с телевизором: внимательно следите за своим самочувствием и немедленно прекращайте работу, если оно начинает устойчиво ухудшаться.

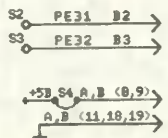
Теперь о магнитофоне. Лучше, если их будет два; оба — с дистанционным управлением работой двигателя и низкого (не выше третьего) класса (высококачественные магнитофоны, как ни странно, хуже воспроизводят сигнал БК!). Зачем нужен второй магнитофон? При обменах программами чужие кассеты часто не читаются из-за несоответствия установке головки вашего магнитофона; ее приходится перенастраивать, потом возвращаться к «своей» установке... В итоге многократного выполнения этих операций вы станете обладателем набора программ, для считывания каждой из которых нужно особо регулировать головку.

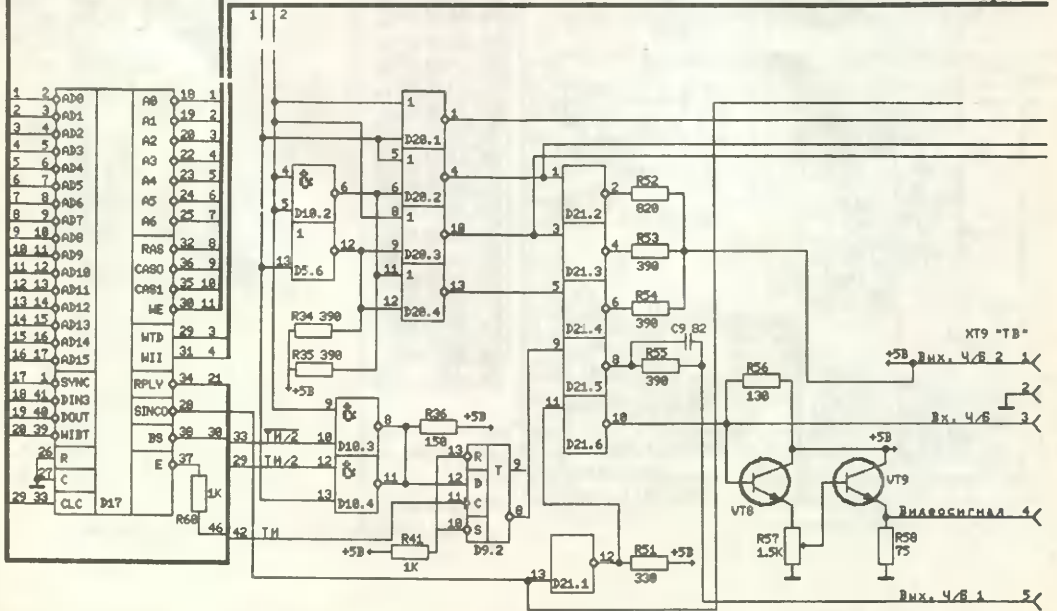
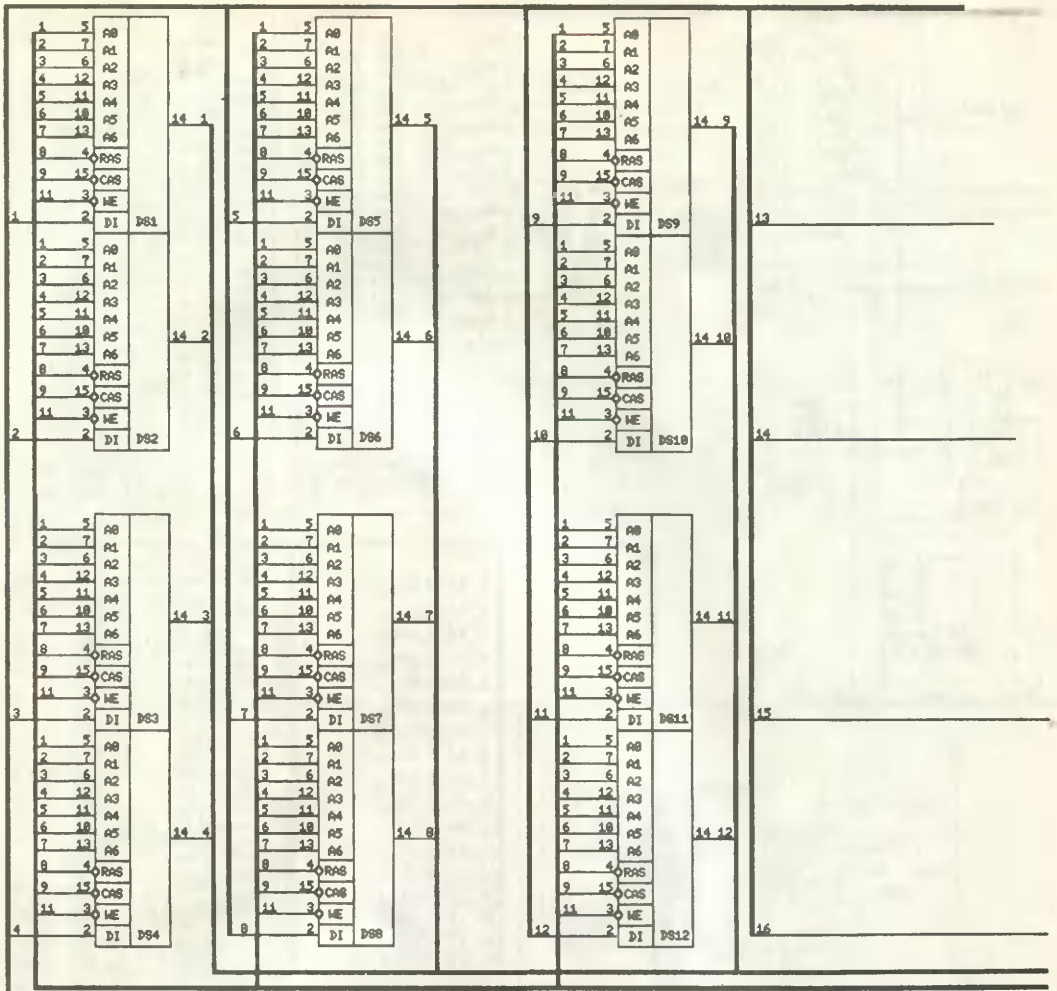
При наличии же двух магнитофонов один используется только для своих кассет, а другой — для чужих; для упрощения перенастройки его головки удобно заменить винт регулировки ее положения на другой, более длинный, торчащий из корпуса, что позволит

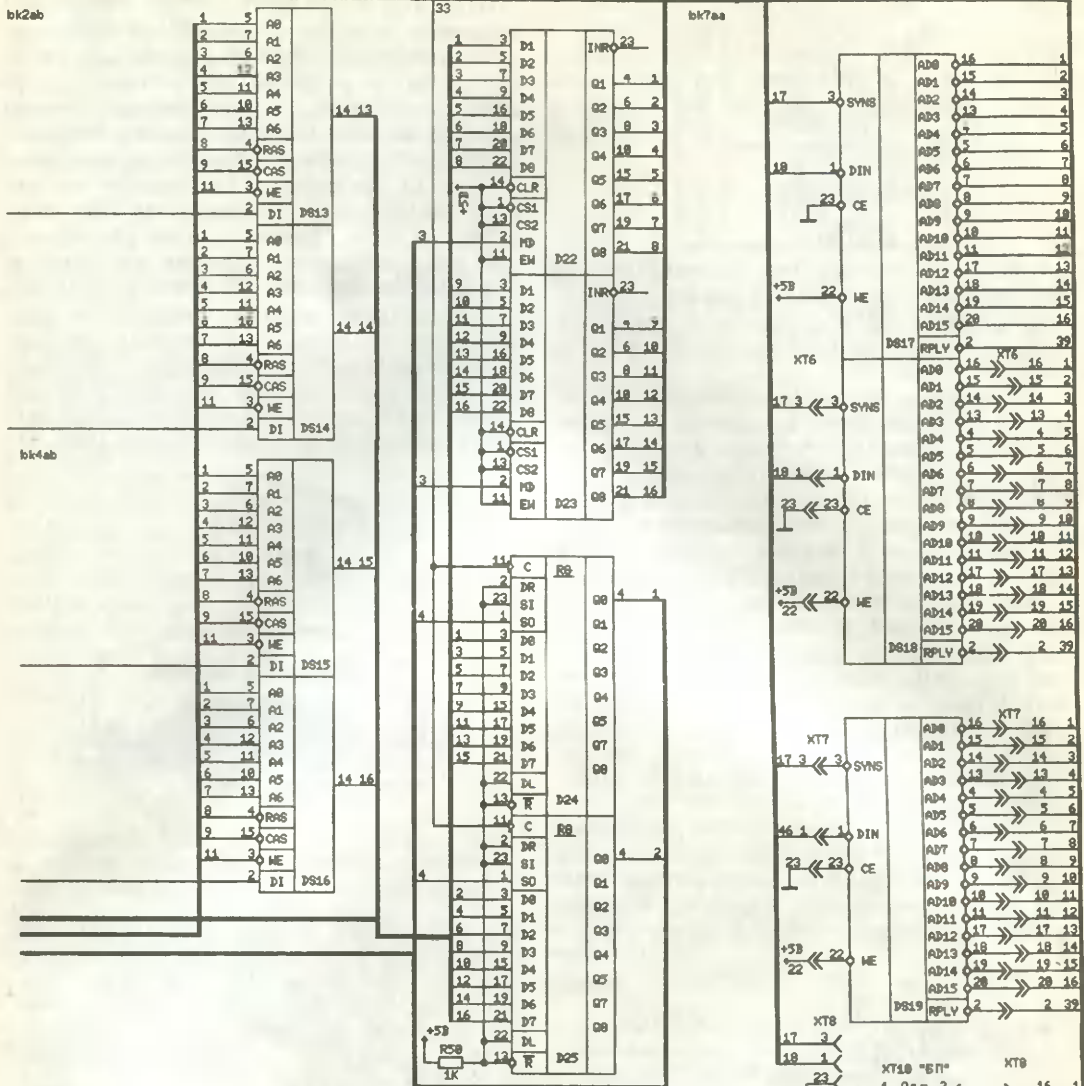




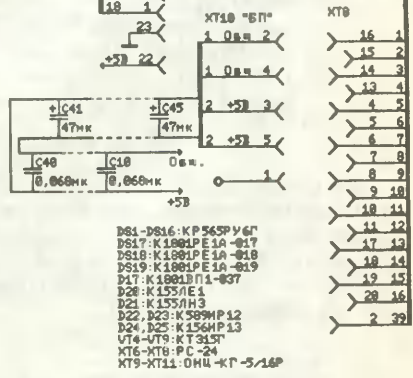
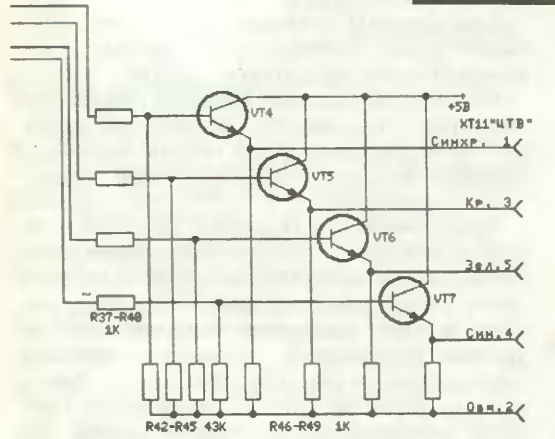
- D1: K555AH1
D2: K561ПУ4
D3, D9: K155ТМ2
D4: K1801ВП1-014
D5: K555AH2
D6: K561AE5
D7: K155/A3
D8: K531B91
D10: K155/A13
D11, D13: K155HP1
D12: K531AП21
D14: K1801ВН1А
D15-D16: K589HP12
- VD1-VD6: K8521A
VT1, VT2: KT245T
VT3: KT3180T
KT1: P5-16
KT2: P6-16
KT3: KTS СМП5В-64,94x 9В-23-В
KT4: ОМЧ-КР-5,16P







77



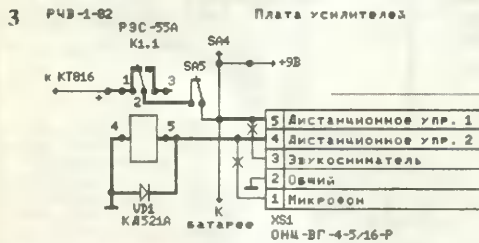
- D81 - D816: KP565P V6T
- D817: K1801PE1A - 817
- D818: K1801PE1A - 818
- D819: K1801PE1A - 819
- D17: K1801D11 - 807
- D28: K1551E1
- D21: K1551H3
- D22, D23: K589MP12
- D24, D25: K1564P13
- V74 - V79: K-T315T
- XT6 - XT8: PC-24
- XT9 - XT11: 01M - K-T-5, 16P

| Разводка цепей питания | | |
|------------------------|--------------------------------------|----------|
| Номинал | Устройство | Ком-такт |
| +5В | D2, E1, E2, E4-E8 | 1 |
| | D81-D84 | 8 |
| | D1, D3, D5-D7, D9-D11, D13, D20, D21 | 14 |
| | D6, D12, K71, K72 | 16 |
| | D15-D18, D22-D25, DS17-DS19, K78 | 24 |
| | D4, D14, D19 | 42 |
| | E3 | |
| Общий | D1, D3, D5-D7, D9-D11, D13, D20, D21 | 1 |
| | D2, D6, D12, K71, K72 | 8 |
| | D15-D18, D22-D25, DS17-DS19, K78 | 7 |
| | D81-D84 | 14 |
| | D4, D14, D19 | 16 |
| | | 24 |
| | | 42 |

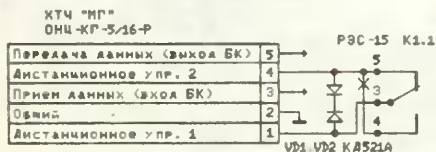
регулировать головку без применения отвертки, просто пальцами. В корпусе, естественно, нужно сделать прорезь, чтобы новый винт не мешал перемещению головки вперед-назад.

Удивительно, но очень немногими пользователями БК осознается удобство дистанционного управления двигателем магнитофона. Впрочем, зачастую такая возможность в магнитофоне просто отсутствует. Между тем дистанционное управление чрезвычайно удобно: компьютер сам в нужные моменты включает и выключает двигатель магнитофона, а человеку остается лишь подготавливать соответствующий режим работы. Если же дать возможность компьютеру переключать еще и режимы «запись — воспроизведение — перемотка», а также определять (хотя бы приблизительно), какое место ленты находится перед головкой... Но об этом как-нибудь после. Сперва — дистанционное управление двигателем.

Сделать его очень просто. На рис. 3 приведено одно из возможных решений этой проблемы для магнитофона «Легенда-404». Аналогично может быть доработан и любой другой магнитофон.



Эта схема рассчитана на тех, кто не хочет пользоваться специальной заглушкой для отключения в магнитофоне дистанционного управления. Им придется переделать включение реле РС-15 на процессорной плате БК на обратное (рис. 4).



Извините, что заставил вас еще раз вскрывать компьютер; чтобы не повторять этого, подключите заодно параллельно гнездам 2 и 3 разъема «МГ» конденсатор емкостью 0,1 мкФ. Это существенно улучшит качество записи программ. Опытные пользователи подбирают для разных типов лент различные RC-фильтры и подключают их по мере необходимости, однако это под силу далеко не всем. Приведенная же рекомендация дает достаточно хороший результат в большинстве случаев.

Многолетняя практика подсказала еще один простой прием, значительно увеличивающий надежность считывания длинных программ: в магнитофонную кассету позади прижимной планки с фетровой подушкой вводится небольшой кусочек поролона (рис. 5).



5

Это практически полностью устраняет продольные колебания ленты, и процесс считывания идет без сбоя даже при сильно изношенных магнитных головках. Лучше провести подобную доработку для всех используемых кассет сразу, чтобы потом случайно не попасть в просак.

К сожалению, этот прием несколько увеличивает износ головки, однако выгоды от его использования превышают потери.

Следующий важный момент — юстировка магнитной головки. Без объяснений понятно, что правильность положения головки — главный параметр, определяющий надежность считывания.

Существуют два основных способа юстировки: на слух и с помощью специальных тестирующих программ. Первый сводится к такой регулировке головки, чтобы при воспроизведении программы были максимальны уровень громкости и слышимость высоких, «звонящих» звуков. При известном навыке и остроте слуха он дает хорошие результаты. Более надежен, однако, второй способ. Наилучшей программой, разработанной для его

реализации, является TESTMAG С. Гуторенко (ИНФО. 1988. № 5). Она выводит на экран специфическую дугорбую кривую, вид которой зависит от качества юстировки и служит руководством к действию. Работа сводится к такой установке головки, при которой расстояние между вершинами горбов

максимально, а их форма близка к идеальной. Юстировка проводится быстро и качественно.

Имеющийся в системном обеспечении БК внутренний тест магнитофона (тест № 5), к сожалению, неуклюж и отнимает много времени.

Окончание следует.

Многоголосие на БК-0010

Приводимая программа наверняка доставит немало приятных минут начинающим и опытным пользователям БК: она открывает ранее неизвестные музыкальные возможности и позволяет проанализировать работу встроенного таймера, недостаточно хорошо известного рядовым пользователям. Может быть, эта программа вдохновит кого-нибудь на создание многоголосых мелодий на этом от природы одноголосом компьютере, тем более что для этого не потребуются никаких аппаратных доработок.

Вот текст программы:

```
1000A
```

```
12737,140,177706,12737,40,177712,12700,
4,12701,2000,5341,3402,77003,770,13711,
177710,52737,100,177716,42737,100,
177716,757
```

Она засылает в ячейку 1016 значение, определяющее число звуковых псевдоканалов; в приводимом варианте их 4, однако диапазон засылаемых значений простирается от 1 до 340. Управлять звуковой октавой можно, меняя содержание ячейки 1002, а ячейка 1010 может принимать значения 20, 40, 100, 200, 60, 140, 300, 160, 340, 360. Регистр 177706 определяет диапазон счетчика 177710, а регистр 177712 управляет скоростью работы таймера.

С. Ивашинников

Калейдоскоп

А. ИВАНОВ (г. Свердловск) предлагает команду FIND «имя.ASC» в программном режиме заменять операторами OPEN «имя. ASC» FOR INPUT и CLOSE;

перераспределять память с помощью программируемых ключей:

```
10 KEY1, "CLEAR200, адр." + CHR$(10)
20 KEY2, "DEL10-20" + CHR$(10)
```

```
...
```

```
...
```

```
NN ?"Нажмите клавиши AP2/1.AP"/2
```

```
и AP2/5
```

```
NN+1 END
```

Здесь «адр.» — адрес свободной области.

А. БОКОВ (Москва) предлагает для Фокала осуществлять ветвление программ по выбору, делая аргументом оператора G имя переменной, например:

```
1.02 S K(1)=1.1:S K(2)=2.2:S
K(3)=3.3:...K(10)=10.1
1.03 A "Введите номер N",N
1.04 G K(N)
1.05 Q
1.10 T !"ПЕРВЫЙ
1.19 Q
2.20 T !"ВТОРОЙ
2.29 Q
3.30 T !"ТРЕТИЙ
3.39 Q
...
10.10 T !"ДЕСЯТЫЙ
10.19 Q
```

П. ЛУЧКА (пос. Новотошковский, Ворошиловградская обл.) предлагает переходить из Бейсика в монитор без нарушения программ и определений функциональных клавиш по командам: DEF USR0=&O100400 и A\$ =USR0(A\$), могущим исполняться и программно. Возврат в Бейсик — по клавише СТОП.

Б. ПОЛТОРАНИН (г. Ташкент) предлагает процедуру восстановления исходного состояния экрана проводить программным образом, используя следующую подпрограмму:

```
19.10 S X=FX(1,40):I (X)19.2,19.5;
D 19.3:G 19.5
19.20 I (X+1)19.4
19.30 X FCHR(155)
19.40 X FCHR(157)
19.50 S X=FX(1,44):I (X)19.55,19.7;
D 19.6:G 19.7
19.55 I (X+1)19.65
19.60 X FCHR(159)
19.65 X FCHR(156)
19.70 S X=FX(1,56):I (-X)19.8:R
19.80 X FCHR(154):R
```

А. ЛАНЕЕВ (г. Ташкент) прислал сразу четыре предложения. Первое — способ убирать служебную строку и вводить в нее сообщения:

```
10 F0KE &0160, &040000+M
20 ?".....
```

Здесь М — номер позиции в служебной строке.

А вот процедура однократной прокрутки экранной страницы:

```
10 INPUT "Скорость прокрутки":N
20 CLS
30 .....
40 .....
50 .....
60 .....
70 .....
80 .....
90 .....
100 FOR P=-5416 TO -5160
110 POKE -76,P
120 FOR R=1 TO N
130 NEXT R,P
140 END
```

Между строками 20 и 100 производится прорисовка экранной страницы (число соответствующих строк произвольно). В строке 140 вместо END может стоять RETURN если процедура реализована как подпрограмма.

Возвратиться в Бейсик из монитора без потери текста программы можно по команде
? S123222 <ВВОД>

И наконец, имитация оператора PLAY для БК.

```
10 FOR J=1 TO 10
20 READ R
30 FOR A=1 TO 20/R+8 'задание
                               темпа проигрывания
40 POKE &0177716,64
50 FOR E=1 TO R
60 NEXT E
70 POKE &0177716,144
80 NEXT A,J
90 DATA 1,2,3,4,5,6,7,8,9,1 'задание
                               нот (1-9)
```

Появившееся в продаже координатное устройство ввода («мышь») УВК-01 заинтересовало многих владельцев БК-0010. В редакцию начали приходить вопросы о его использовании. Сегодня консультант И. Панченков рассказывает о том, как подключить УВК-01 к БК.

Прежде всего нужно вставить вилку УВК-01 в разъем УП БК, затем включить БК. На экране монитора появится отклик

```
?00 AT 0.00
ГОТОВНОСТЬ К РАБОТЕ
*■
```

Теперь нужно нажать одновременно клавиши AP2 и ; и отпустить их. На экране появится отклик

*■

Нажмите ЛАТ ЗАГЛ Р _ Т ВВОД;¹ появится отклик +; нажмите РУС Т С; появится отклик\$. Теперь введите 1264 АИ... 127000... и далее все коды программы, приведенной на с. 6 описания УВК-01. Не обращайтесь внимания на числа, появляющиеся на экране после ввода запятой.

После ввода всех кодов программы нажмите СТОП, ЛАТ, введите 1264G. Если программа введена правильно, на экране появится отклик + и установится связь между УВК-01, БК и монитором. Теперь, перемещая УВК-01 по столу, можно рисовать на экране.

Если же отклика не появится, нужно проверить правильность ввода программы, для чего нажать СТОП, СБРОС, а затем одновременно AP2 СУ У. Затем нажмите РУС, введите 1264 АИ и далее, нажимая клавишу «запятая», шаг за шагом проверьте правильность всех введенных кодов. При нахождении ошибки исправьте ее и повторите запуск программы (СТОП, ЛАТ, 1264G).

Правильно введенную программу запишите на магнитную ленту: она является драйвером УВК-01 и пригодится впоследствии при работе с графическими редакторами. Ее данные: адрес загрузки 1264, длина 320.

Сейчас уже есть мощные графические редакторы, в которые этот драйвер входит составной частью, например редактор P01M (автор А. Бакерин, Москва). Появляются и игровые программы с управлением от этой «мыши». Не забывайте, что использование «мыши» вне соответствующих программ практически бессмысленно.

П о п р а в к а. В ИНФО № 2 в третью снизу строку в левом столбце на с. 52 вкралась ошибка. Программный текст в ней следует читать так:

```
А0=""+А0+"0"
```

Э. КАРАВАЕВА

методист Черниговского института усовершенствования учителей

Нетрадиционные формы обучения на уроках информатики

Изучение курса ОИВТ в большинстве ПТУ Черниговской области осуществляется в безмашинном варианте.

Специфика предмета заключается в большом объеме новых понятий и терминов. Как в этих условиях добиться качественного выполнения учебных программ, сознательного усвоения изучаемого материала, повысить уровень алгоритмической культуры учащихся?

На помощь пришли нетрадиционные формы и методы проведения учебных занятий. О ряде из них мы узнали со страниц журнала «Информатика и образование». Обобщили и распространили свой опыт. На уроках применяем различные виды упражнений, связанные с получаемой профессией учащихся, самостоятельные работы в форме кроссвордов и мини-кроссвордов. Используется программа контроля знаний на ПМК, разработанная преподавателем ПТУ № 5 А. С. Савченко. При разработке алгоритмов широко применяются игровой способ, магнитные диски, опорные листки.

Необычно начинает урок на тему «Регистровая память» старший преподаватель С. Г. Мурашко ПТУ № 1. Преподаватель сообщает, что кроме уже известной учащимся стековой памяти в ПМК есть еще одна независимая память и сегодня они должны научиться использовать ее при вычислениях. Обращаясь к группе, он спрашивает: «Может, кто-нибудь из вас знает, как называется эта память?» Учащиеся затрудняются ответить на этот вопрос. Преподаватель предлагает учащимся сделать это совместными усилиями, заполнив пустые клетки названия темы предстоящего урока, заранее записанной на доске.

Слово «память» в нижние клетки записывает преподаватель, а верхние клетки заполняют ученики, поочередно выбирая карточки и отвечая на поставленные вопросы. Отвечая на второй, третий, четвертый и пятый вопро-

| ТЕМА УРОКА | | | | | | | | | | |
|------------|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 4 | 5 |
| | | | | | | | | | | |
| | | | | | | | | | | |

81

сы, учащиеся записывают только первую букву ответа в клеточку с соответствующим индексом, а отвечая на первый вопрос, учащиеся записывают семь букв ответа в клеточки с индексом 1. Чтобы учащиеся не угадали название памяти, надо сделать так, чтобы карточка с индексом 1 оказалась последней. Итак, учащиеся решили «мини-кроссворд», повторили при этом ранее изученный материал, и в результате создан положительный эмоциональный настрой.

Карточки

№ 2. Геометрическая фигура, используемая в блок-схеме алгоритма, обозначающая начало и конец алгоритма (*овал*).

№ 3. Составная команда, начинающаяся со служебного слова ЕСЛИ, называется командой (*ветвления*).

№ 4. Понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение указанной цели или на решение поставленной задачи (*алгоритм*).

№ 5. Система обозначений и правил для единообразной и точной записи алгоритмов и их исполнения (*язык*).

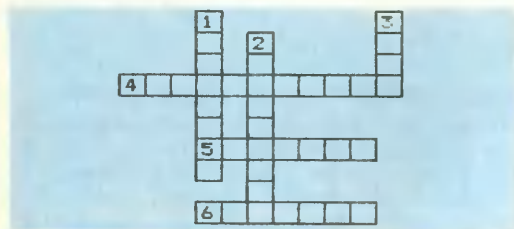
№ 1. Электронное устройство, предназначенное для кратковременного запоминания и хранения многоразрядного числа (*регистр*).

В журнале «Информатика и образование» (1989, № 3) в статье «Компьютерный кроссворд в учебных целях» предложен кроссворд на тему «Принципы устройства ЭВМ» для ПЭВМ. Наши преподаватели применяют этот

кроссворд в безмашинном варианте для отработки понятий и терминов по данной теме. В ПТУ № 15 Т. Б. Гриц по материалам этой статьи разработала программу кроссворда для КУВТ-2 «Ямаха», который используется в учебном процессе.

Преподаватели информатики ПТУ № 16 А. Ю. Караваев и ПТУ № 34 В. Н. Кресан разрабатывают и применяют учебные кроссворды, привлекают к этой творческой работе учащихся. Например, кроссворды 1 и 2 применялись на уроке общественного смотря знаний.

Кроссворд 1



По горизонтали. 4. Наука, изучающая законы и методы накопления и переработки информации. 5. Английский математик, который ввел в 1936—37 гг. концепцию абстрактной «вычислительной машины». 6. Язык программирования высокого уровня.

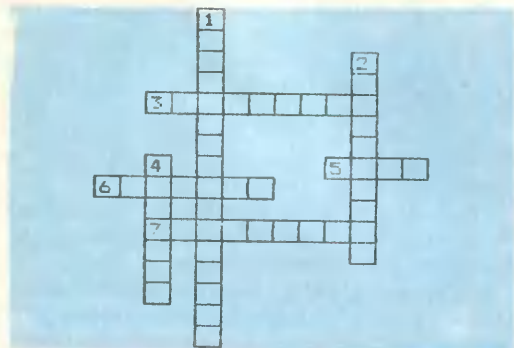
По вертикали. 1. Универсальный способ решения ряда подобных задач. 2. Принятое в научно-популярной и научной литературе название ЭВМ. 3. Служебное слово алгоритмического языка.

Ответы

По горизонтали. 4. Информатика. 5. Тьюринг. 6. Фортран.

По вертикали. 1. Алгоритм. 2. Компьютер. 3. ПОКА.

Кроссворд 2



По горизонтали. 3. Последовательность вводимых в ЭВМ операций и команд, необ-

ходимых для решения задачи. 5. Единица количества информации. 6. Периферийное устройство ЭВМ. 7. Центральное устройство ЭВМ, обрабатывающее информацию.

По вертикали. 4. Устройство, обеспечивающее диалог между микро-ЭВМ и человеком. 1. Периферийное устройство ЭВМ. 2. Устройство ввода информации в ЭВМ.

Ответы

По горизонтали. 3. Программа. 5. Байт. 6. Принтер. 7. Процессор.

По вертикали. 4. Дисплей. 1. Графопостроитель. 2. Клавиатура.

Сознательному усвоению терминологии, понятий алгоритмического языка и умению им пользоваться способствует применение «магнитной» доски. Служебные слова алгоритмического языка четко записаны на небольших планшетах из картона, на обратной стороне планшетов приклеены небольшие магнитики. Наборы таких планшетов находятся перед доской в кабинете. Откидные доски классной доски покрыты жестью, что позволяет быстро крепить планшеты. Перед уроком преподаватель записывает пару алгоритмов на откидных досках, служебные слова алгоритмического языка помещает не все (некоторые умышленно опускает, другие ставит неправильно). Вызванные учащиеся быстро наводят порядок в записи алгоритмов (мелом писать не надо, вытирать тоже).

Элементы соревнования позволяют внести в урок опросные листы. Методики их применения просты: на каждом ученическом столе два стандартных листа, на одном записаны 10—15 вопросов, а на другом ответы, но ответы под другими номерами, причем ответов на два-три больше, на некоторые (один-два вопроса) ответа нет. Учащиеся одного ряда задают несколько вопросов, учащиеся другого ряда зачитывают ответы с листа. Затем меняются ролями.

Можно использовать эти листы и для проверки знаний. Учащиеся отвечают кратко в тетрадях, например: 1—5; 2—8 и т. д. (первая цифра — вопрос, вторая — ответ). Затем можно включить водоскоп и спроцировать на доску правильные ответы. Учащиеся проверяют работы друг у друга, ставят оценки. Об оценках договариваются в начале работы (ПТУ № 12, преподаватель Н. Н. Малышенко).

Применение нетрадиционных методов и форм на уроке информатики позволяет управлять познавательной деятельностью учащихся, поддерживать интерес и внимание, оперативно устанавливать обратную связь, учить творчески и качественно.

Компьютер, цвет, язык

Так назвал С. Фадеев статью, опубликованную в ИНФО, № 4 за 1989 г. Мы давно занимаемся этой проблемой, но подходим к ней с нескольких иных позиций, а именно: что дает проблема звукоцвета для изучения самого языка. Свои результаты в этой области мы и предлагаем вниманию читателей журнала.

Один из родоначальников французского символизма А. Рембо писал:

*А — черно, Е — бело, У — зелено,
И — ярко-красное,
О — небесного цвета!*

*Вот так, что ни день,
что ни час,*

*Ваши скрытые свойства беру я
на цвет и на глаз.*

*Вас на цвет и на запах
я пробую, глосные.*

Вот так четко была сформулирована связь между звуками речи (гласными) и цветом. Позднее этой проблемой занимались во Франции К. Нироп, в Германии — А. Шлегель. Обращался к этому вопросу и Р. Якобсон.

Многое в исследовании проблемы звукоцвета сделал в последние годы А. П. Журавлев (см. его книги: Фонетическое значение. Л.: Изд-во. ЛГУ, 1974. С. 50—53; Диалог с компьютером. М.; Молодая гвардия, 1987. С. 124—152).

В настоящее время получило распространение утверждение, что «хроматическая цветовая гамма связана с гласными звуками — они цветные. Согласные располагаются по черно-белой оси: от светлых зубных... к темным губным» (Энциклопедический словарь юного филолога. М.: Педагогика, 1984. С. 107).

А. П. Журавлев, анализируя данные многочисленных экспериментов, приводит таблицу звукоцветовых соответствий (термин «звукобуква» принадлежит А. П. Журавлеву).

Как видно из табл. 1, у А. П. Журавлева другие звукоцветовые соответствия по сравнению с А. Рембо. Здесь, на наш взгляд, сказывается особенность национально-языкового мышления: по всей вероятности, оценка звукоцветовых соответствий различна у представителей разных языков.

А. П. Журавлев считает, что ударные гласные при определении звукоцвета удваиваются. Кроме того, он вводит понятие двух-

цветной звукобуквы. Например Ё — ее количество разделяется поровну между О и Е. Синева Й выражена слабо, поэтому количество Й сокращается наполовину и приплюсовывается к И.

Таблица 1

| Звуко-буквы | Цвета | Звуко-буквы | Цвета |
|-------------|---|-------------|---|
| А Я О | густо-красный ярко-красный светло-желтый или белый | И Й У | синий синеватый темно-синий, темно-лиловый, темно-синезеленый |
| Е Ё | зеленый желто-зеленый | Ю Ы | голубоватый темно-коричневый или черный |

83

Указанные факты натолкнули нас на мысль проанализировать цветовую гамму цветковых слов и цветных описаний с помощью компьютера. С этой целью была подготовлена программа на Бейсике для ДВК-2М. В связи с тем что экран монитора у компьютера монохромный, выходные данные о цвете выдаются в виде словесного описания.

Программа составлена так, что она учитывает «цвет» каждого гласного звука в тексте. После ввода в ЭВМ всего текста программа «просчитывает» эту информацию и выдает результат в виде названия цветов в порядке их убывания. Так как каждый гласный звук имеет свой «цвет», то вполне естественно, что любой отрезок текста оказывается многоцветным. Определяющими мы считали первые три цвета, так как именно они, по нашему мнению, создают основной цветовой фон.

Размышляя над проблемами звукоцвета, мы предположили, что в случае действительного наличия звукоцветовых соответствий, четче всего они должны быть представлены в словах — названиях цветов. Поэтому в компьютер мы ввели цветковые слова: синий, зеленый, желтый, красный.

Вот как определил компьютер «цвет» этих

слов:

синий синий, синеватый
зеленый зеленый, желто-зеленый
желтый желто-зеленый, светло-желтый
красный густо-красный, темный

Для объективности выводов мы решили проверить на «цвет» отдельные грамматические формы приведенных выше прилагательных. Анализ дал следующие результаты:

синее зеленый, синий
синяя ярко-красный, синий
желтая густо-красный, желто-зеленый
зеленая густо-красный, зеленый
красное густо-красный, белый

Итак, слово «синий» — синего цвета, а слово «синяя» — ярко-красного! В чем здесь дело?

84

Теперь мы обратили внимание на то, что, говоря о звукоцветовых соответствиях, А. П. Журавлев приводит буквы. И термин использует «осторожный» — **звукобуква!** Однако цветовую гамму создают звуки, а буквы, как условное отражение звуков, цвета не имеют. Они могут приобретать цвет лишь условно, в плане методическом, как об этом рассказывал в своей статье С. Фадеев.

Мы внесли уточнения в утверждения А. П. Журавлева и соответственно в нашу программу для компьютера. Учли явление редукции гласных (сокращения гласных в безударном положении): О в первом предударном после твердых согласных произносится как А; А, Е в той же позиции после мягких согласных как И (мы говорим «вАда», «чИсы» и т. д.). Были введены в программу редуцированные (ослабленные безударные звуки: Ъ как средний между Ы и А — компьютер обозначал его через & и Ь как средний между Е и И — компьютер обозначал его через #.

Мы уже говорили, что, по мнению А. П. Журавлева, при звукосчете ударные гласные удваиваются. Развивая это положение, мы решили, что неударные И, Ы, У, а также А (на месте О) равны 1. А величину редуцированных Ъ и Ь мы приняли за 0,5.

По новой программе для слов «красная», «синяя», «синее» и т. д. мы получили такие результаты:

красная густо-красный
синяя синий, синеватый
желтая желто-зеленый
зеленая желто-зеленый, зеленый

Теперь мы убедились, что наша гипотеза верна: в звуковом облике слова, называемом цвет, отражается цвет, создаваемый составляющими его звуками.

Возник еще один вопрос. Подтверждается ли такое машинное «видение» звука человеческим восприятием?

Чтобы решить это, мы провели следующий эксперимент.

На одном из потоков 1-го курса факультета иностранных языков мы предложили студентам отрывок из стихотворения С. Есенина «Кузнец»:

*Душно в кузнице угрюмой,
И тяжел несносный жар.
И от визга и от шума
В голове стоит угар.
К наковальне наклоняясь,
Машут руки кузнеца.
Сетью красной рассыпаясь,
Вьются искры у лица.
Взор отважный и суровый
Блещет радугой огней,
Словно взмах орла, готовый
Унести за даль морей.*

Мы попросили студентов представить нарисованную поэтом картину в цвете и указать три наиболее распространенных цвета, которые они видят в есенинском тексте.

Затем текст был введен в ЭВМ. Вот что сообщила машина (ниже приводится листинг. Текст — затранскрибированный; обозначены не гласные буквы, а гласные звуки, которые произносятся).

```
ДУ'ШНѢ В КУ'ЗНИЦѢ УГРЮ'МЪЙ  
И ТИЖО'Л НИСНО'СНЫЙ ЖА'Р  
И АТ ВИ'ЗГѢ И АТ ШУ'МѢ  
В ГѢЛ'ВѢ СТАИ'Т УГА'Р  
К НѢКАВА'ЛНѢ НѢКЛАНА'ЙСѢ  
МА'ШУТ РУ'КИ КУЗНИЦА'  
СЕ'ТЮ КРА'СНѢЙ РѢССЫПА'ЙСѢ  
ВЬУ'ТСѢ И'СКРЫ У ЛИЦА'  
ВЗО'Р АТВА'ЖНЫЙ И СУРОВО'Й  
БЛЕ'ЩИТ РА'ДУГѢЙ АГНЕ'Й  
СЛОВ'НО ВЗМА'Х ОРЛА' ГОТОВО'Й  
УНИСТИ'СЬ ЗА ДА'ЛЬ МОРЕ'Й
```

ЦВЕТА ДАННОГО ТЕКСТА ПРИВЕДЕНЫ В ПОРЯДКЕ

УБЫВАНИЯ:

ГУСТО-КРАСНЫЙ
СИНИЙ
ТЕМНО-СИНИЙ, ТЕМНЫЙ, СИНЕ-ЗЕЛЕНый, ТЕМНО-ЛИЛОВый
СИНЕВАТЫЙ
СВЕТЛО-ЖЕЛТЫЙ ИЛИ БЕЛЫЙ
ЗЕЛЕНый
МРАЧНЫЙ ТЕМНО-КОРИЧНЕВый ИЛИ ЧЕРНЫЙ

Сравним данные человека и компьютера (см. табл. 2).

Итак, имеется ряд интересных моментов.

Во-первых, у человека наблюдается увеличение разброса цветов от первого к третьему.

Во-вторых, все так или иначе отметили красный цвет. Он отмечался некоторыми и как первый, и как второй, и как третий.

Сопоставляя «человеческие» и машинные данные, можно в целом говорить о значи-

Таблица 2

| Порядок цвета | Цвет | Человек | Компьютер |
|---------------|--------------|---------|-----------------------------|
| Первый | красный | 68 % | густо-красный |
| Второй | серо-черный | 24 % | синий |
| | красный | 40 % | |
| Третий | серо-черный | 37 % | темно-синий, темный |
| | сине-голубой | 40 % | |
| | серо-черный | 22 % | сине-зеленый, темно-лиловый |

тельном совпадении цветового восприятия человека и ЭВМ. Учитывая, что цветовой компонент устанавливался только на основе «цветовой» окраски гласных звуков, мы думаем, что общая теория звукоцвета применительно к языку имеет под собой определенное основание.

Надо обратить внимание на одно обстоятельство. Машина не отмечает серый, серо-черный цвета; лишь в гамме третьих цветов отмечает «темный». Дело, очевидно, в следующем: ЭВМ анализировала только гласные звуки и устанавливала цвет на этой основе, человек рисовал себе целостную картину. Мы уже отмечали в начале статьи, что согласные звуки создают черно-белый фон. Есть этот фон и в человеческом восприятии текста. Машина же пока этого не видела.

Мы составили еще один вариант программы, куда был заложен фон, создаваемый согласными звуками. Как и авторы Энциклопедического словаря юного филолога, мы исходили из того, что светлый фон создается зубными согласными С, Ц, Т, Д и т. д.), а темный — губными (Б, В, М) и заднеязычными (Г, К, Х).

В анализируемом тексте около 70 % «светлых» согласных и примерно 30 % — «темных». Фон, таким образом, серый. Его-то и видели студенты, отмечая серый цвет.

Когда отработала программа, учитывающая цветовые значения и гласных, и фон, создаваемый согласными звуками, то получилась такая картина:

ЦВЕТА ДАННОГО ТЕКСТА ПРИВЕДЕНЫ В ПОРЯДКЕ
УБЫВАНИЯ:
НА СВЕТЛО-СЕРОМ ФОНЕ:
ГУСТО-КРАСНЫЙ
СИНИЙ
ТЕМНО-СИНИЙ, ТЕМНЫЙ, СИНЕ-ЗЕЛЕНый, ТЕМНО-ЛИЛОВый
СИНЕВАТЫЙ
СВЕТЛО-ЖЕЛТЫЙ ИЛИ БЕЛЫЙ
ЗЕЛЕНый
МРАЧНЫЙ ТЕМНО-КОРИЧНЕВый ИЛИ ЧЕРНЫЙ

нова «Весна», в котором для описания картины весны используются цветовые свойства звуков.

*Сиинь соон сийй селле соонг се
Сиинд сеельф сиййк сигналъ сеель
Лиий левиш ляак льяйсильлюк
Ляай луглет ляав лилийн лед...*

Компьютер так «увидел» в цветах эту картину весны:

СИИ'НЬ САО'Н СИИ'Й СИЛЛЕ' САО'НГ СЕ'
СИИ'НД СИЙЕ'ЛФ СИИ'ИИК СИГНА'ЛЪ СИЙЕ'ЛЪ
ЛИИ'Й ЛИВИ'Ш ЛЯА'К ЛЬЯИСИЛЬ'У'К
ЛИА'Й ЛУГЛЕ'Т ЛИА'В ЛИЛИИ'Н ЛЕ'Д

ЦВЕТА ДАННОГО ТЕКСТА ПРИВЕДЕНЫ В ПОРЯДКЕ
УБЫВАНИЯ:

НА СЕРОМ ФОНЕ:
СИНИЙ
ЗЕЛЕНый
ГУСТО-КРАСНЫЙ
СИНЕВАТЫЙ
СВЕТЛО-ЖЕЛТЫЙ ИЛИ БЕЛЫЙ
ТЕМНО-СИНИЙ, ТЕМНЫЙ, СИНЕ-ЗЕЛЕНый, ТЕМНО-ЛИЛОВый

85

Наверное, можно согласиться с таким видением весны!

Вернемся, однако, к «цвету» слов, обозначающих цвет.

Читатель, наверное, обратил внимание, что из семи цветов спектра мы упомянули только о четырех. Оставшиеся три слова требуют особого разговора. Меньшего — слово «голубой». ЭВМ, анализируя звукокомплекс ГЪЛУБОЙ, отмечает такую структуру «цвета» этого слова: б-л-ый, синева-тый, темно-синий. Можно, наверное, согласиться с таким толкованием: в целом получается «голубой» = смягченный синий.

Оранжевый и фиолетовый — слова иноязычного происхождения. Поэтому прогонка их через компьютер с «русской» звуковой программой дала, как и следовало ожидать, вполне бессмысленный ответ: оранжевый — густо-красный, черный; фиолетовый — густо-красный, зеленый.

Пришлось взяться за этимологический словарь и провести через ЭВМ все этимологические цепочки. Среди них мы нашли у слова «оранжевый» немецкую параллель — orange, а у слова «фиолетовый» — латинскую — viola.

Учитывая особенности звучания этих слов в немецком и латинском языках, мы заложили их в компьютер. Результат:

оранжевый — красный, желтый, белый
фиолетовый — белый, красный, синий
В словаре русского языка С. И. Ожегова, к примеру *о р а н ж е в ы й* — это желтый с красноватым оттенком, а *ф и о л е т о в ы й* — синий с красным оттенком.

Мы заинтересовались вопросом, как выглядят «цвета» слов, обозначающих составные цвета. Компьютер выдал такие данные:

Для убедительности мы заложили в ЭВМ стихотворение поэта-символиста А. Туфа-

КРАВА'ВЫЙ

ЦВЕТА ДАННОГО ТЕКСТА ПРИВЕДЕНЫ В ПОРЯДКЕ
УБЫВАНИЯ:

ГУСТО-КРАСНЫЙ
СИНЕВАТЫЙ
МРАЧНЫЙ ТЕМНО-КОРИЧНЕВЫЙ ИЛИ ЧЕРНЫЙ
СИНИЙ

БАГРА'ННЫЙ

ЦВЕТА ДАННОГО ТЕКСТА ПРИВЕДЕНЫ В ПОРЯДКЕ
УБЫВАНИЯ:

ГУСТО-КРАСНЫЙ
СИНЕВАТЫЙ
МРАЧНЫЙ ТЕМНО-КОРИЧНЕВЫЙ ИЛИ ЧЕРНЫЙ
СИНИЙ

СУ'МРАЧНЫЙ

ЦВЕТА ДАННОГО ТЕКСТА ПРИВЕДЕНЫ В ПОРЯДКЕ
УБЫВАНИЯ:
ТЕМНО-СИНИЙ, ТЕМНЫЙ, СИНЕ-ЗЕЛЕНый, ТЕМНО-ЛИПОВЫЙ
МРАЧНЫЙ ТЕМНО-КОРИЧНЕВЫЙ ИЛИ ЧЕРНЫЙ
СИНЕВАТЫЙ

СИНИЙ
ГУСТО-КРАСНЫЙ

ПА'СМУРНЫЙ

ЦВЕТА ДАННОГО ТЕКСТА ПРИВЕДЕНЫ В ПОРЯДКЕ
УБЫВАНИЯ:

ГУСТО-КРАСНЫЙ
СИНЕВАТЫЙ
ТЕМНО-СИНИЙ, ТЕМНЫЙ, СИНЕ-ЗЕЛЕНый, ТЕМНО-ЛИПОВЫЙ
МРАЧНЫЙ ТЕМНО-КОРИЧНЕВЫЙ ИЛИ ЧЕРНЫЙ
СИНИЙ

Из всего сказанного напрашивается вывод, что обработка языкового материала на ЭВМ в ряде случаев дает вполне реальные и интересные результаты, помогает подтвердить правильность тех положений, которые до сих пор воспринимались как вольность, ибо были отмечены лишь поэтическим мышлением. Компьютер позволяет по-новому взглянуть на язык, забраться в такие глубины, которые до сих пор были недоступны человеческому взору.

Ж. КАРАЕВ, С. РАХ

Использование компьютеров в казахской школе

Компьютеризация школы — одно из ключевых звеньев осуществляемой реформы народного образования. Недавно созданная концепция национальной казахской общеобразовательной школы уделяет этой серьезной проблеме немало внимания. Специфическая особенность национальных школ — перегрузка учебного плана на 4—5 часов по сравнению с русскоязычными. Это объясняется обязательным изучением трех языков: национального, русского и иностранного. Первоочередной задачей становится определение целей, места и границ применения ЭВМ в школе. Мы эту проблему рассматриваем в контексте чисто специфических условий Казахстана, в котором насчитывается огромное количество школ с казахским языком обучения.

Применению компьютеров для повышения эффективности учебно-воспитательного процесса, и прежде всего для индивидуализации и активизации познавательной деятельности школьников, посвящена экспериментальная и исследовательская работа, проводимая в НИИ педагогических наук им. Алтынсарина на базе национальных школ Алма-Аты.

Обучение с помощью компьютера стимулирует интерес учащихся к учебным предметам. Обучающая программа должна предусматривать, по возможности, как разнообразие заданий, так и поисковую деятельность учащегося для получения правильного результата, что развивает познавательную активность школьников. Кроме того, работа с компьютером снимает боязнь учащихся перед неверным ответом, который в обычных условиях вызвал бы бурную реакцию одноклассников. Это положение целиком подтверждают данные эксперимента по изучению русского языка в старших классах казахской школы. Ранее замкнутые учащиеся проявляют активность в работе с компьютером и при достаточной отработке навыков смело вступают в диалог. Развитию умения вести беседу и поддерживать диалог на неродном языке способствует включение в обучающую программу заданий по работе со связным текстом. В данном случае наблюдается активизация общения между учениками, между учителем и учеником.

Работа в компьютерном классе позволяет с помощью локальной сети активно вовлечь каждого ученика в учебный процесс

и организовать обучение в индивидуальном режиме. Создание протоколов результатов выполнения работы с указанием затраченного времени, количества допущенных ошибок, их типов и т. п. немаловажно, поскольку позволяет учителю скорректировать подход к каждому ученику, в случае необходимости оценить работу каждого школьника непосредственно на уроке. Средствами компьютерной техники удалось существенно дифференцировать диалог со школьниками в зависимости от успешности выполнения ими заданий. Наш алгоритм обучения построен так, что продолжение обучения разрешается только при выполненном правильно задании с разбором или указанием ошибок. Такая дифференциация способствует овладению программой каждым учеником, причем наблюдается рост качества знаний и экономия учебного времени при выполнении программы. Например, учебным планом обязательно выделены часы на закрепление и повторение пройденного материала. Пройденный эксперимент показал, что с помощью локальных сетей можно почти вдвое сократить эти часы и качественно улучшить уроки, насытив их интересным материалом и новыми заданиями.

Это в какой-то мере решает одну из проблем казахских школ — снятие перегрузки учащихся.

Остановимся подробнее на результатах проводимого нами эксперимента по применению компьютеров в учебном процессе казахской школы на уроках русского языка. Практика показала, что возможности компьютера в изучении этого предмета исключительно велики. Появилась возможность увеличить объем необходимой тренировки, без которой, как подчеркивается А. А. Ле-

онтьевым в его концепции обучения предметам языкового цикла, невозможно полноценное формирование навыков. Компьютер дает возможность оптимизировать темп чтения, развивает умение письменного общения, обеспечивает более глубокое понимание языкового материала. Анализ результатов эксперимента свидетельствует о более высокой успеваемости школьников экспериментальных групп, применявших при изучении курса русского языка персональные компьютеры. Подтверждением этого явилось повышение количества правильных ответов на предложенные вопросы учеников экспериментальных групп по сравнению с контрольными.

Улучшение качества знаний мы связываем с рядом факторов: с получением за единицу учебного времени большего объема информации, выполнением большего количества упражнений, формированием у учащихся положительной мотивации, интереса к изучаемой дисциплине. Подтверждением эффективного использования ЭВМ выступают и результаты анкетирования, собеседования с учителями и школьниками.

Результаты экспериментальных работ дают возможность утверждать, что использование компьютеров позволяет сделать обучение более гибким: организовать переход от фронтального к индивидуальному, от него — к групповому и т. п. В обучении русскому языку (при работе с компьютерами) хорошо зарекомендовала себя групповая форма обучения. В этом случае члены группы, чаще всего два ученика, обсуждая между собой пути достижения цели, активно общаются, дискутируют на неродном языке.

Таким образом, в национальной школе компьютеры несомненно помогут повысить качество обучения.

87

Хоть гвозди забивай

Удар с ускорением 40 g, купание, температуру от -47°C до $+63^{\circ}\text{C}$, понижение давления воздуха до величины, соответствующей высоте 4,5 км над уровнем моря, и другие воздействия может выдерживать, по утверждению компании Paravant Computer Systems, портативный компьютер RLT-88. Основное запоминающее устройство этого компьютера, естественно, не диски, а платы памяти на микросхемах, емкость которых может достигать 8М байт.

Вес компьютера с аккумуляторами — 5,4 кг; о цене, к сожалению, не сообщается.

Компьютерный Mighty Mouse

Его зовут «Powermouse 100», и это действительно могучая мышь. Только представьте себе — на ней 40 кнопок! «Клавиатура на колесиках», производимая компанией Prohance Technologies Inc., предназначена для пользователей пакета Lotus 1-2-3. Кнопки вызывают выполнение наиболее употребляемых макрокоманд, а также позволяют вводить числовые данные.

Хотя это устройство стоит по американским меркам недорого — 195 долларов, вряд ли оно станет популярным. Вы согласны?

ЧТО?
МОЖЕТ?
ЭВМ

Данная публикация — первая из серии, в которой будут разобраны задачи повышенной сложности и конкретизирован подход к созданию олимпиадных задач, которого придерживается коллектив во главе с академиком *Н. Н. Красовским*.

«Уральские» задачи

Задача о прибыли

Эта статья дает пример учебной задачи о рациональном выборе действий. Задача предлагалась для олимпиады по информатике «Урал-ИНФО-89». В ней проявляются черты, характерные для практических проблем оптимизации. Автор обсуждает вариант решения (не обязательно самый лучший). Более подробный разбор вариантов решения будет приведен в сборнике учебных задач, подготовленном к изданию сотрудниками Института математики и механики УрО АН СССР.

Постановка задачи

Некто имеет листы металла в форме квадрата. Некто умеет делать коробочки следующим образом. Пусть сторона квадрата равна a . По углам квадрата вырезаются квадраты со стороной b . Из крестообразной фигуры свертывается и сваривается коробочка. Ее можно продать. При этом остаются еще четыре квадрата со стороной b каждый. Их тоже можно продать или сделать из них коробочки. Последние также можно продать. Останется 16 квадратов. И т. д. В распоряжении Некто довольно много различных «можно». Надо отобрать те из них, которые выгоднее.

Например, Некто, зная условия купли-продажи, может решить: всего выгоднее ему выполнить три этапа работы. На первом этапе вырезать квадраты размером a_1 , на втором — размером a_2 , на третьем — размером a_3 и, стало быть, произвести и продать $1+4+16$ коробочек и 64 остаточных квадрата.

Условия торговли таковы.

Цена r квадратного листа пропорциональна его площади s , т. е.

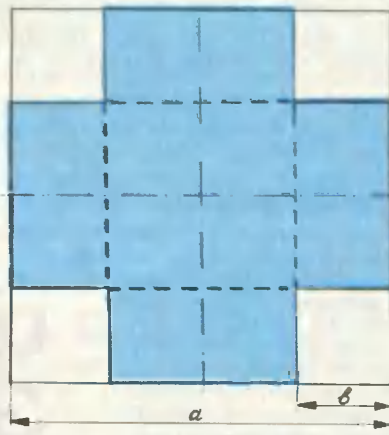
$$r = qs. \quad (1.1)$$

Цена h коробочки пропорциональна ее объему v , т. е.

$$h = pv. \quad (1.2)$$

Затраты t на изготовление коробочки пропорциональны длинам швов и линий изгибов. Имеется в виду, что это затраты на разрезание, сваривание и изгибание. В связи с этим затраты на изготовление коробочки высотой x из квадрата размером y определяются формулой

$$t = l(y - 2x) + mx. \quad (1.3)$$



Стало быть, доход u от такой коробочки определяется равенством

$$u = pv - l(y - 2x) - mx. \quad (1.4)$$

В данных формулах размеры выражаются в сантиметрах, цены — в копейках. Коэффициент p положителен, коэффициенты q , l , m неотрицательны.

Не будем обсуждать достоверность этих формул. Примем, что таковы условия рынка и производства. Формулы выбраны попроще, чтобы не заслонять выкладками суть дела. Можно даже сказать, что приводимые ниже расценки являются сказочными.

Спрашивается: сколько этапов надо осуществить и какие квадраты вырезать на каждом этапе, чтобы суммарный доход E от продажи коробочек и металла получился максимальным?

Подчеркнем, что задача допускает оптимальную организацию производства лишь в пределах описанной выше схемы, т. е. допустимо только подбирать количество этапов и размеры квадратов, вырезаемых на каждом этапе. Другие возможности оптимального раскроя не должны рассматриваться. Например, не допускается сначала раскроить лист на квадраты самых разнообразных размеров (впрочем, постановка задачи не исключает одного варианта такого раскроя, когда квадраты получаются последовательным делением сторон пополам). При желании читатель может сам сочинить различные обобщения задачи о раскрое, не вызывая себя описанной схемой производства.

Решить предложенную задачу аналитически, т. е. в виде формулы, трудно. (Если ограничить производство одним этапом, то задачу решить аналитически не слишком трудно. Такие задачи решают в школе. Однако если не ограничивать допустимое число этапов, то трудности аналитического решения с увеличением числа этапов сильно возрастают. Впрочем, читатель, который любит аналитические выкладки, может попытаться строить решение, опираясь на них.)

Предлагается составить программу численного решения задачи на компьютере. Программа должна запрашивать исходные данные p, q, l, m, a ; после их ввода компьютер должен вычислить оптимальное количество этапов n , оптимальные значения a_1, a_2, \dots, a_n для сторон квадратов, вырезаемых на каждом этапе, доход E , выдать результаты вычислений.

Например, программа, составленная автором, при исходных данных

$$p = 0.1, q = 0.015, l = 0.2, m = 0.2, a = 250 \quad (1.5)$$

выдала результат

$$n = 3, a_1 = 43.50, a_2 = 7.50, a_3 = 1.50, E = 117973.363 \quad (1.6)$$

Приведем возможное решение задачи. Как и в большинстве случаев работы с компьютером, данный способ решения — лишь один из многих возможных. Более того, этот способ далек от совершенства. Зато его можно пояснить без больших хлопот.

Квадратный лист со стороной y будем обозначать символом $л(y)$. Максимальное количество денег, которое получим, если из квадрата $л(y)$ изготовим коробочки за оптимальное число этапов и на каждом этапе будем вырезать квадраты оптимальным образом, назовем максимальным доходом от квадрата $л(y)$ и будем обозначать $g(y)$.

Пусть по ходу производства к началу какого-то этапа получается лист $л(y)$. Пусть мы решаем на этом этапе вырезать из листа $л(y)$ квадраты $л(x)$. Размер x пока не предполагается оптимальным. Имеем в виду, что

$$0 \leq x \leq y/2. \quad (2.1)$$

(При $x = y/2$ коробочка не получается, но формально удобно допустить и значение $x = y/2$, полагая, что получается коробочка нулевого объема. Также значение $x = 0$ удобно допустить формально. Но изготовление при $x = 0$ в счет этапов мы не включаем. При $x = 0$ производство завершается на квадрате $л(y)$.)

Предположим, что на следующем этапе, который начинается с квадратов $л(x)$, будем уже действовать оптимально. От нашего квадрата $л(y)$, таким образом, получим доход

$$f(y, x) = p(y - 2x)^2 x - l(y - 2x) - mx + 4g(x). \quad (2.2)$$

Равенство (2.2) для дохода $f(y, x)$ справедливо при $x > 0$. Но мы можем не делать коробочек из квадрата $л(y)$, а сразу его продать. В этом случае $x = 0$. Получим

$$f(y, 0) = qy^2. \quad (2.3)$$

Итак, по смыслу максимального дохода g имеем равенство

$$g(y) = \max_{0 \leq x \leq y/2} f(y, x). \quad (2.4)$$

Запись $\max_{0 \leq x \leq y/2} (y, x)$ означает, что при

фиксированном значении y выбирается такое значение x из отрезка $0 \leq x \leq y/2$, при котором величина $f(y, x)$ достигает максимума. Из (2.2), (2.3) и (2.4) вытекает своеобразное уравнение

$$g(y) = \max \left(\max_{0 \leq x \leq y/2} (px(y - 2x)^2 - l(y - 2x) - mx + 4g(x)), qy^2 \right) \quad (2.5)$$

для функции g . Правую часть (2.5) следует понимать так. Перебираются все значения x из интервала $0 < x \leq y/2$. Если всякий раз при этих x величина $f(y, x)$ (2.2) получается не больше, чем qy^2 , то

$$g(y) = qy^2. \quad (2.6)$$

Иначе выбираем

$$g(y) = \max f(y, x) \quad (2.7)$$

при $0 < x \leq y/2$. Школьник может спросить: а как быть, если при некоторых x из интервала $0 < x \leq y/2$ получается $f(y, x) > qy^2$, но функция $f(y, x)$ на этом интервале максимума не достигает? И может ли такое быть? Пусть подумает над этим вопросом сам. Или подождет до следующего параграфа.

Очевидно, справедливо равенство

$$g(0) = 0 \quad (2.8)$$

Уравнение (2.5) получилось как необходимое условие для максимального дохода g (при условии, что решение задачи о максимальном доходе существует). Но тогда возникают вопросы. Достаточно ли найти как-либо решение $g(y)$ уравнения (2.5) при условии (2.8), чтобы это решение было максимальным доходом? Этот вопрос связан с вопросом: является ли решение $g(y)$, $y \geq 0$ уравнения (2.5) при условии (2.8) единственным? Можно продолжить сомнения, спрашивая: имеет ли вообще исходная задача решение? И т. д. Можно предложить читателю самому найти ответы на такие вопросы.

А мы попробуем пока найти как-нибудь функцию $g(y)$, $y \geq 0$.

Максимальный доход

Надо бы уметь находить величину $g(y)$ для любого действительного числа y из отрезка $0 \leq y \leq a/2$. Но всех таких чисел очень много. Поступим обычным для практики способом. Примем, что стороны тех квадратов, которые возникают по ходу дела, кратны некоторому малому числу $d > 0$. Например, это может получаться из-за особенностей работы режущего инструмента.

Итак, будем решать задачу (2.5) о максимальном доходе $g(y)$ численно для каждого значения

$$y_j = dj, \quad j = 0, 1, 2, \dots, 0 \leq y_j \leq a/2 \quad (3.1)$$

прямым перебором всех нужных значений

$$x_k = dk, \quad k = 0, 1, 2, \dots, 0 \leq x_k \leq y_j/2. \quad (3.2)$$

Поскольку теперь мы ограничиваемся лишь числами y_j и x_k вида (3.1), (3.2), то исчезают сомнения в существовании максимума, высказанные в предыдущем параграфе. Правда, теперь возникнут новые сомнения. Например, можно ли, выбирая число $d > 0$ достаточно малым, приблизиться к решению идеальной задачи, в которой допускаются все действительные числа y и x ? Школьник может обдумать это сам.

Предлагаемый алгоритм отражает эволюцию величины $g(y)$ при возрастании переменной y от $y=0$ до $y=a/2$. Стало быть,

этот алгоритм будет вычислять последовательно величины $g(y_j)$ в порядке возрастания чисел y_j (3.1).

Согласно (2.8) полагаем $g(0) = 0$. Идем по индукции. Пусть значения $g(y)$ уже вычислены для всех $y = y_s$ при $s = 0, 1, \dots, j$. Тогда согласно равенству (2.5) величина $g(y_{j+1})$ вычисляется из условия

$$g(y_{j+1}) = \max_{0 \leq x_k \leq y_{j+1}/2} (px_k(y_{j+1} - 2x_k)^2 - l(y_{j+1} - 2x_k) - mx_k + 4g(x_k)), \quad qy_{j+1}^2. \quad (3.3)$$

Максимум в правой части (3.3) вычисляется при фиксированном значении y_{j+1} по всем допустимым x_k .

Так мы получим набор чисел

$$g(y_0), g(y_1), \dots, g(y_u), \quad u = a/(2d). \quad (3.4)$$

Оптимальные значения a_i

Зная величины $g(y_j)$, можно найти те квадраты $l(a)$, которые надлежит вырезать в оптимальном производстве.

Полагаем $a_0 = a$. Идем по индукции. Пусть число a_i уже найдено. Тогда число a_{i+1} находится из решения следующей задачи:

$$f(a_i, a_{i+1}) = \max_{0 \leq x_k \leq a_i/2} f(a_i, x_k) = \max_{0 \leq x_k \leq a_i/2} (px_k(a_i - 2x_k)^2 - l(a_i - 2x_k) - mx_k + 4g(x_k)), \quad qa_i^2. \quad (4.1)$$

Задачи (4.1) решаем численно для каждого значения a_i перебором всех нужных чисел x_k .

Вычисления заканчиваются, когда впервые получается $a_{i+1} = 0$. Значение i и будет равно числу n этапов в оптимальном производстве.

Ниже приведен алгоритм вычислений, формализованный в виде программы на Бейсике. Результаты, приведенные в конце первого параграфа, получены как раз этой программой, причем был выбран шаг $d = 0.25$.

Программа

```

10 *** БЛОК ВВОДА НАЧАЛЬНЫХ ДАННЫХ ***
20 *
30 * КОМАНДА ОЧИСТКИ ЭКРАНА
40 CLS
50 * ПЕЧАТЬ ИМЕНИ ПРОГРАММЫ
60 PRINT "          ДОХОД"
70 * ПРОПУСТИТЬ ДВЕ СТРОКИ НА ЭКРАНЕ
80 PRINT:PRINT
85 * ВВОД ИСХОДНЫХ ДАННЫХ
90 PRINT "INPUT : "
100 PRINT "КАКОВЫ "
110 INPUT "  ЦЕНА ЕДИНИЦЫ ОБЪЕМА P=":P
120 INPUT "  ЦЕНА ЕДИНИЦЫ ПЛОЩАДИ Q=":Q
130 INPUT "  КОЭФФИЦИЕНТЫ
      ТРУДОВЫХ ЗАТРАТ L,M=":L,M
140 INPUT "  НАЧАЛЬНЫЙ РАЗМЕР
      ЛИСТА A=":A
150 INPUT "  ШАГ ВЫЧИСЛЕНИЙ      D=":D
160 PRINT:PRINT
170 *

```

```

180 '*****
190 '
290 'ОЦЕНКА КОЛИЧЕСТВА ШАГОВ
      РЕЖУШЕГО ИНСТРУМЕНТА
300 U=INT(A/(2*J)+5)
305 'ОБЪЯВЛЕНИЕ РАЗМЕРА МАССИВА
310 DIM B(U)
320 '-----
400 '
410 'КОМАНДЫ 500-550
415 'ВЫЧИСЛЯЮТ ПЕРЕБОРОМ
420 'МАКСИМАЛЬНОЕ ЗНАЧЕНИЕ B(J)
430 '
500 FOR J=0 TO U
505 Y=D*J
507 B(J)=Q*Y*Y
510 FOR K=0 TO J/2+0.1
515 X=D*J
520 Z=(F*(Y-2*X)^2*X-L*(Y-2*X)-M*X)*
      SGN(K)+4*J(K)
530 IF B(J)<Z THEN B(J)=Z
540 NEXT K
550 NEXT J
600 '
680 '-----
685 PRINT "ВЫЧИСЛЕНИЕ ЗНАЧЕНИЙ A(I)
690 PRINT "ДЛЯ СТОРОН ТЕХ КВАДРАТОВ,
700 PRINT "КОТОРЫЕ НАДО ВЫРЕЗАТЬ ПРИ
710 PRINT "ОПТИМАЛЬНОМ СПОСОБЕ РАБОТЫ
711 PRINT "РАБОТЫ (ПЕРЕБОРОМ)
800 '
810 '-----
830 'ОБЪЯВЛЕНИЕ РАЗМЕРА МАССИВА A(I)
831 'С ЗАПАСОМ
835 N1=INT(LOG(A/(2*J))/LOG(2))
840 DIM A(N1)
850 '
860 A(0)=A
890 FOR I=0 TO N1
900 Z=Q*A(I)*A(I)
905 A(I+1)=0
910 FOR J=0 TO A(I)/(2*J)+0.1
920 W=(F*(A(I)-2*J*J)^2*J*J-
      L*(A(I)-2*J*J)-M*J*J)*SGN(J)+
      4*J(J)
930 IF W>Z THEN Z=W: A(I+1)=D*J
940 NEXT J
950 ПРОВЕРКА УСЛОВИЯ ОКОНЧАНИЯ
960 IF A(I+1)=0 THEN 990
970 PRINT "A(";I+1;)"=";
975 PRINT USING "#####.####";
      A(I+1)
980 NEXT I
990 N=I
1000 PRINT "КОЛИЧЕСТВО ЭТАПОВ РАБОТЫ
      N=":N
1010 '
1020 GOSUB 1220
1030 '-----
1035 PRINT
1040 PRINT "ЦЕНЫ ПРОДУКЦИИ ПО ЭТАПАМ "
1045 PRINT "-----"
1046 PRINT
1050 S=0
1055 IF N=0 THEN 1058 ELSE 1060
1058 PRINT "ВЫГОДНЕЕ ПРОДАТЬ ЦЕЛЫЙ
      ЛИСТ " : R=Q*A*A: GOTO 1190
1060 FOR I=0 TO N-1
1070 H=F*(A(I)-2*A(I+1))^2*A(I+1)
1080 PRINT "ЦЕНА КОРОБОЧКИ НА " ;
      I+1; "-ОМ ЭТАПЕ H=";
1085 PRINT USING "#####.####";H
1090 U=H-L*(A(I)-2*A(I+1))-M*A(I+1)

```

```

1100 PRINT "ДОХОД ОТ КОРОБОЧКИ НА " ;
      I+1; "-ОМ ЭТАПЕ U=";
1105 PRINT USING "#####.####";U
1110 W=4^(I)*J
1120 PRINT "ДОХОД ОТ ВСЕХ КОРОБОЧЕК
      НА " ; I+1; "-ОМ ЭТАПЕ W=";
1125 PRINT USING "#####.####";W
1130 S=S+W
1135 PRINT
1140 NEXT I
1150 R=4^N*G(INT(A(N)/D+.5))
1170 PRINT "ДОХОД ОТ ВСЕХ
      КОРОБОЧЕК S=";
1175 PRINT USING "#####.####";S
1180 PRINT "ДОХОД ОТ ПОСЛЕДНИХ
      ОБРЕЗКОВ R=";
1185 PRINT USING "#####.####":R
1190 PRINT "ОБЩИЙ ДОХОД E=G(A);
      E=";
1195 PRINT USING "#####.####";S+R
1200 PRINT "ПРИБЫЛЬ V=E-Q*A*A; V=";
1205 PRINT USING "#####.####";
      S+R-Q*A*A
1210 STOP
1215 '-----
1220 PRINT:PRINT "НАЖМИТЕ ЛЮБУЮ
      КЛАВИШУ " : I$=INPUT$(1):RETURN

```

Примечания

Выше отмечено, что описанный алгоритм, а стало быть, и программа далеки от совершенства. Алгоритм можно улучшать. Школьник может формализовать алгоритм по своему вкусу, в частности выбирая для программы более приятный и удобный для него язык. Можно ввести запрос о желаемой точности вычислений; тогда целесообразно вычислять в режиме итераций. И т. д.

Задачу можно усложнять. Можно предусмотреть перенастройку программы на другие формулы стоимостей. Например, цену коробочки можно определить формулой

$$h = (p + p_1 \times \exp(-p_2(1/3 - x/(y + 0.0001))^2)(y - 2x)^2 x. \quad (6.1)$$

Добавочный член, зависящий от параметров $p_1 > 0$ и $p_2 > 0$, показывает, что отдается предпочтение коробочкам, форма которых близка к кубу.

Можно обобщить задачу: не задавая исходного размера листа, оговорить только начальный капитал. На этот капитал можно купить то или иное количество квадратных листов (в зависимости от их размера) и организовать изготовление коробочек так, чтобы получить наибольшую прибыль. Тогда программа должна еще подсказывать, листы какого размера a выгоднее всего покупать. И т. д.

Если предусмотреть перенастройку программы на формулы стоимостей, отличные от (1.1) — (1.3), то появятся понятные изменения в программе. Они будут связаны с определением новых функций стоимостей:

DEF FNF (X) = ...,

DEF FNH (Y, X) = ...,
 DEF FNT (Y, X) = ...

Можно модернизировать программу так, что она будет работать в режиме итераций с переменным шагом, причем по ходу итераций вычисления будут сгущаться в окрестности искомого значения a_i и значения $y=0$. Такая программа работает сначала как исходная, с некоторым не слишком малым шагом $d^{(1)}$, и находит первое приближение для значений $E^{(1)}$, $a_i^{(1)}$. Затем шаг $d^{(1)}$ уменьшается в несколько раз и с новым шагом $d^{(2)}$ строится новое разбиение $y_j^{(2)}$, но уже не для всего отрезка $0 \leq y \leq a/2$, а выбираются только такие значения $y_j^{(2)}$, которые лежат на отрезках, окружающих значения $a_i^{(1)}$, и на отрезке, примыкающем к значению $y=0$. Длины этих отрезков равны $zd^{(1)}$, где z не очень большое число. На таком разбиении $y_j^{(2)}$ программа работает опять как исходная и определяет второе приближение $E^{(2)}$, $a_i^{(2)}$. И т. д.

92

Школьник может попробовать составить подобную программу сам. Такая программа, составленная автором, дала для

$$p=10, q=0.000001, l=0.000001, \\ m=0.000001, a=40 \quad (6.4)$$

следующие результаты:

$$n=7, a_1=6.9446, a_2=1.2058, a_3=0.2093, \\ a_4=0.0363, a_5=0.0063, a_6=0.0011, a_7= \\ =0.0002, E=48358.74. \quad (6.5)$$

Исходные данные (6.4) выглядят странно. Но они были выбраны такими, чтобы лучше проследить работу итерационной программы.

При вычислениях по этой программе получилась следующая итерационная последовательность, уточняющая значение E , a_i .

$$d^{(1)}=0.25 \quad n^{(1)}=3 \\ a_1^{(1)}=7.00, a_2^{(1)}=1.25, a_3^{(1)}=0.25 \\ E^{(1)}=48354.9984$$

$$d^{(2)}=0.0625 \quad n^{(2)}=4 \\ a_1^{(2)}=6.9375, a_2^{(2)}=1.1875, a_3^{(2)}=0.1875, \\ a_4^{(2)}=0.0625 \\ E^{(2)}=48358.2578$$

$$d^{(5)}=0.0009 \quad n^{(5)}=6 \\ a_1^{(5)}=6.9443, a_2^{(5)}=1.2060, a_3^{(5)}=0.2090, \\ a_4^{(5)}=0.0361, \\ a_5^{(5)}=0.0059, a_6^{(5)}=0.0009 \\ E^{(5)}=48358.7327$$

$$d^{(6)}=0.0002 \quad n^{(6)}=7 \\ a_1^{(6)}=6.9446, a_2^{(6)}=1.2058, a_3^{(6)}=0.2092, \\ a_4^{(6)}=0.0364, \\ a_5^{(6)}=0.0064, a_6^{(6)}=0.0010, a_7^{(6)}=0.0002 \\ E^{(6)}=48358.7367$$

Для тех же исходных данных p, q, l, m, a вычисления по исходной программе с шагом 0.5 дали следующие результаты:

$$n=2, a_1=7.0, a_2=1.0, E=48320.00. \quad (6.7)$$

Вычисления для тех же исходных данных по той же программе, но с шагом 0.1, дали результаты

$$n=3, a_1=6.9, a_2=1.2, a_3=0.2, E=48356.84. \quad (6.8)$$

Итерирующая программа обладает тем достоинством, что позволяет вычислять оптимальные значения a_i с большой точностью гораздо быстрее, чем с такой же точностью вычисляла бы эти значения исходная программа с фиксированным очень малым шагом d . Но итерирующая программа с переменным шагом обладает и своими недостатками. Если формулы для стоимостей не очень регулярны и начальный шаг $d^{(1)}$ неосторожно выбрать недостаточно малым, то на первой итерации можно потерять значения a_i , близкие к тем оптимальным значениям a_i , которые на самом деле решают задачу. Школьник может сам попробовать построить такой пример. Это можно попробовать сделать уже для стоимости, которая дается формулой (6.1).

Полезно также обратить внимание на одну особенность решения задачи. При выбранных выше исходных данных p, q, l, m, a (6.4), где q, l, m во много раз меньше, чем p , и при существенном уменьшении шага вычислений оптимальное при этом шаге количество этапов производства заметно увеличивается. В то же время оптимальный доход E , начиная с некоторого умеренно малого шага, возрастает несущественно. Более того, можно, например, поставить задачу об оптимальном производстве при условии допустимости не более чем одного этапа. Сравнивая при одинаковых исходных данных (1.5) решение этого варианта, результаты вычислений (1.6) по программе с фиксированным довольно большим шагом $d=0.25$ и результаты работы итерационной программы для исходной задачи, мы увидели бы, что величина оптимального дохода во всех трех случаях примерно одна и та же. Аналогичный вывод получается для весьма разнообразных исходных данных. Поэтому с точки зрения организации производства уточнение результатов $E^{(s)}$, $a_i^{(s)}$ путем уточнения вычислений мало полезно. Может быть, это уточнение даже вредно, так как оно рекомендует усложнение производства при малом выигрыше в доходе E . Это явление связано с тем, что исходные формулы стоимостей примитивны. В частности, формула для дохода не учитывает в явном виде удорожание производства с увеличением числа эта-

пов работы. Однако подобные явления возможны и в серьезных задачах, где формулы стоимостей продукта и затрат на производство выглядят более реалистичными. Но все-таки главной причиной слабого влияния выбираемого количества этапов производства на доход есть следствие пренебрежения дополнительными наценками на продукт за то или иное его качество. Если читатель спросит, зачем же автор демонстрирует теорию на примитивных формулах стоимостей и стоит ли развивать математическую теорию производства, усложняемого грошовой выгоды ради, то ответим: затем, чтобы показать, что математика может лишь обработать экономические факты. Хочешь иметь эффективную экономику, конструируй эффективные цены. Тогда и математика поможет улучшать дело.

Напомним также следующее обстоятельство из области «искусства ради искусства». Математик и вычислитель должен поддерживать форму постоянной тренировки. А для этого выявление малых эффектов часто оказывается наиболее полезным.

Рекомендуем читателю попробовать сочинить такие формулы для расценок, при которых уточнение вычислений сказывается существенно не только на оптимальном числе этапов работы, но и на величине оптимального дохода. Подскажем, что такой может быть формула, которая увеличивает цену коробочки под влиянием следующих факторов: объем коробочки, близость к кубической форме, особенная ценность маленьких коробочек.

Еще одно замечание. В связи с вышесказанным может сложиться впечатление, что имеет место следующее явление. Если выполнять вычисления неограниченное число раз $s=1, 2, \dots$, устремляя шаг $d^{(s)}$ к нулю при $s \rightarrow \infty$, то оптимальное количество $n^{(s)}$ этапов производства будет неограниченно возрастать. Однако в случае, если хотя бы один из коэффициентов q, m не равен нулю, такого явления не возникнет! При стремлении шага $d^{(s)}$ к нулю с ростом s получится такой номер s_0 , начиная с которого оптимальное число n^0 окончательно перестанет возрастать и при всех $s > s_0$ будет $n^{(s)} = n^{(s_0)}$. Это значение $n^{(s_0)}$ и является строго оптимальным числом этапов производства. Разумеется, оптимальные значения $q^{(s)}$ и $E^{(s)}$ при уменьшении шага могут продолжат уточняться и при $s > s_0$ вплоть до исчерпания возможностей используемого компьютера. Школьник может убедиться в этом рассуждением или экспериментально.

Надлежит обратить внимание и на явление противоположного порядка. Пусть числа q, l и

m в формуле стоимости равны 0. Тогда оказывается, что строгое оптимальное решение задачи содержит бесконечное количество этапов и, стало быть, оптимальные размеры a вырезаемых квадратов $l(a)$ составят бесконечную последовательность убывающих положительных чисел. Эту последовательность можно найти, решая задачу аналитически. Школьник может попытаться сделать это сам. На основе такого аналитического решения в программу можно ввести блок, который предусматривает специальные вычисления в случае, когда коэффициенты q, l и m равны 0.

Шутя можно сказать, что описанные выше усовершенствования программы и связанное с ними уточнение результатов дают пример удовлетворения научного любопытства за счет траты интеллектуальных и экономических ресурсов. Однако если говорить серьезно, то выявление и обсуждение подобных явлений увеличивает наши знания, которые могут весьма и весьма пригодиться. А предсказать заранее, что эти знания никогда не пригодятся, или, наоборот, предсказать, где и когда такие знания могут пригодиться, не всегда возможно.

Если предполагать знание основ дифференциального и интегрального исчисления, то можно предложить еще один путь решения. Алгоритм вычисления максимального дохода, приведенный выше, основан на обходе изменения функции $g(y)$ с ростом y . При этом максимизирующее значение x в уравнении (2.5) вычислялось прямым перебором. Но это максимизирующее значение можно вычислять и по правилам дифференциального исчисления. Тогда при определенных условиях можно проследить, как меняется это максимизирующее значение $x(y)$ с ростом y . Таким образом, получаются дифференциальные уравнения, которые описывают изменения максимального дохода $g(y)$ и максимизирующего значения $x(y)$ с возрастанием аргумента y . Численное интегрирование этих уравнений позволяет найти функцию $g(y)$, $0 \leq y \leq a$ и вычислить оптимальные размеры a вырезаемых квадратов.

Заключение

Описанные способы решения относятся к методу, который принято называть динамическим программированием. Можно предложить для ознакомления с этим методом книгу Р. Беллмана «Динамическое программирование». Она издана в СССР в 1960 г., однако мы ссылаемся на нее, так как развитие динамического программирования в большей степени обязано работам Р. Белл-

мана. В настоящее время динамическому программированию посвящена обширная литература. Заинтересованный читатель может сам найти библиографию этого предмета.

В предложенном тексте задано довольно много вопросов. Некоторые намеренно остав-

лены без ответов. Имеется в виду, что читаемое полезно подвергать сомнению, находить в нем хорошее и недостатки, улучшать решения, уточнять и обобщать задачи.

Академик **Н. КРАСОВСКИЙ**

Новое в учебном процессе

В НПО «Комплекс» освоен выпуск учебных робототехнических комплексов:

- ◆ многоординатный учебный робот, управляемый персональной ЭВМ ДВК;
- ◆ пневматический позиционный учебный робот с техническим зрением. Робототехнические комплексы незаменимы в школах, профтехучилищах, техникумах и институтах для обучения основам информатики, программирования, робототехники и ЧПУ.

173003 Новгород, ул. Дмитриевская, 20. Тел. 7-50-88, 98-306.

Автор-2

Инструментальная система
для диалоговой разработки авторских обучающих систем
и тренажеров непрограммистами

Разработка авторских курсов и тренажеров в диалоге.

Неограниченная вложенность директориев (каталогов).

Автоматический синтез текстов.

Грамматический разбор высказываний.

Обработка статистики в информационно-поисковой подсистеме.

Подключение модулей пользователя и фирменных пакетов без привлечения программиста.

Интерактивная графика средствами графических пакетов.

Индивидуальная поставка и сопровождение.

Цены (на май 1990 г., включая стоимость услуг), руб.:

| | ОС типа RT-11, RSX | ОС типа MS DOS |
|---|-----------------------|-------------------|
| Инструментальная обучающая система | 1900 | 2450 |
| Информационный интерфейс (синтез текстов, грамматический анализ высказываний) | 1200 | 1700 |
| Программный препроцессор непрограммирующего пользователя ПЭВМ | | 850 |

Заказы (с реквизитами организации и телефоном) направлять по адресу:
105835 ГСП Москва Е-250, Красноказарменная, 14, МЭИ, Межвузовский центр
(МЦАНИ), Черных А. В. Тел. 233-39-48 (с 8.00 до 23.00).

Проблемы информатизации: взгляд из педвуза

В феврале 1990 г. на базе Минского педагогического института им. А. М. Горького состоялось выездное заседание секции «Подготовка и повышение квалификации педагогических кадров» Общесоюзного научно-методического совета информатизации образования Гособразования СССР.

Рассмотрев на заседаниях ряд наиболее актуальных для современного этапа информатизации образования вопросов, секция приняла рекомендации, которые могут быть использованы при принятии практических решений, а также в ходе подготовки к III пленуму ОНМС Гособразования СССР.

Центры новых информационных технологий (ЦНИТ)

Введение новых информационных технологий (НИТ) в сферу образования привело к появлению пестрого рынка программно-методических средств учебного и управленческого назначения. В качестве источников выступают как государственные структуры (вузы, подразделения АН СССР и АПН СССР и др.), так и кооперативы, во многих случаях использующие потенциал наиболее квалифицированной части разработчиков. Поставщики программно-методической продукции образуют весьма неоднородный конгломерат — по представительству, уровню квалификации, отношению к сфере образования и ведомственной принадлежности. Многие из них к тому же основывают свою деятельность исключительно на коммерческом подходе, отдавая предпочтение вопросам бизнеса перед научно-методическими проблемами информатизации образования. Среди созданных в последнее время государственных ЦНИТ есть центры, работающие на школьное образование и подготовку педкадров, т. е. в сфере деятельности секции ОНМС, например, республиканский

(РСФСР) центр НИТО при Омском пединституте, республиканский (УССР) центр при Главном вычислительном центре Минобразования УССР и др. В сложившейся ситуации естественной реакцией государственной системы образования является инициализация подведомственных научно-методических центров, осуществляющих функции координации, разработки и экспертизы качества программно-методической продукции. В этих условиях создание различных типов ЦНИТ, действующих под эгидой Общесоюзного ЦНИТ (ОЦНИТ) (приказ Гособразования СССР № 86 от 04.12.89 г. «О центрах новых информационных технологий и использовании закупленного импортного продукта») является своевременным и прогрессивным шагом в развитии инфраструктуры информатизации образования.

Вместе с тем функционирование системы ЦНИТ зависит от статуса каждого центра и порядка взаимодействия между ними (в том числе правовых аспектов).

К сожалению, в упомянутом приказе Гособразования СССР и в утвержденном этим приказом типовом Положении о ЦНИТ проявляется опасная тенденция закрепить бюрократические, административные методы организации и руководства деятельностью ЦНИТ со стороны Гособразования СССР. В основу инфраструктуры ЦНИТ закладывается жесткая централизация, предусматривающая: а) создание ЦНИТ совместными приказами с ОЦНИТ (вместо этого достаточно ввести принцип регистрации вновь создаваемых ЦНИТ); излишнее в данном случае требование издавать совместный приказ является не более чем бюрократическим крючком, который способен только сдерживать создание новых ЦНИТ; б) незаконное присвоение Гособразованием СССР общенародной собственности в виде заку-

паемых за народные средства импортной техники и импортного программного продукта и установление принципа «отработки» ЦНИТами низшего уровня за выделяемую им импортную технику и программное обеспечение; закупленная техника и программный продукт являются общей собственностью всех подразделений Гособразования (в том числе и вузов, на базе которых создаются ЦНИТ), поэтому чем быстрее и проще новые ресурсы достигнут конкретных разработчиков в системе Гособразования, тем лучше; в Положении Гособразования нет четких указаний на экономические основы равноправного сотрудничества центров, а между тем именно равноправное (с экономической точки зрения) сотрудничество центров разного уровня является единственно приемлемым принципом, на основе которого в условиях широкого применения хозрасчетных основ создания ЦНИТ возможно обеспечивать стимулы для разработчиков программного обеспечения.

Отмеченные особенности указанного приказа и Положения должны быть поправлены последующими организационными мерами Гособразования в части создания ЦНИТ и организации их работы.

Существенным недостатком этих документов является также отсутствие четкого выделения и ориентации на «отраслевые» ЦНИТ. Например, не ясно, какие ЦНИТ должны действовать в сфере подготовки педкадров, школьного образования и т. п. На базе педвузов рекомендован только один ЦНИТ (Омский пединститут). Похоже, что приказ был недостаточно хорошо подготовлен, и в этом отношении кажется странным, что подготовка такой важности документов ведется без учета мнения квалифицированных экспертов, сосредоточенных в секциях ОНМС.

Секция считает необходимым внести предложения по включению в перечень рекомендованных вузов для создания при них хозрасчетных ЦНИТ с ориентацией на систему просвещения и подготовку педкадров (при этом существенным является не абстрактное пожелание иметь ЦНИТ при том или ином вузе, а наличие в них соответствующих специалистов и организаторов, способных поставить дело нужным образом, — как правило, это члены секций НМС).

Экспертиза учебно-методических и программных средств

Отмечая роль, которую выполняет экспертиза учебно-методических и программных средств, ориентированных на новые ин-

формационные технологии (УМС НИТ), секция считает, что при разработке положения об экспертизе и сертификации программных средств необходимо учитывать следующее.

1. Для обеспечения экспертизы и сертификации УМС НИТ в системе народного образования (СНО) СССР создается сеть региональных (республиканских) экспертных комиссий (РЭК).

Главной целью ее создания является обеспечение высокого качества используемых в учреждениях СНО УМС НИТ.

2. Сеть РЭК образуется в рамках структуры создаваемой сети региональных (республиканских) НИТ (РЦНИТ).

3. В качестве основного рабочего органа экспертизы УМС НИТ должны выступать экспертные комиссии, функционирующие при РЦНИТ, которые получают статус региональных (республиканских) экспертных комиссий.

4. РЭК регистрируются в ОЦНИТ и наделяются правами рекомендации УМС НИТ к использованию в учебных заведениях СНО в своих регионах (республиках).

5. Состав РЭК формируется по предложению РЦНИТ из числа ведущих специалистов в области НИТ СНО и других отраслей народного хозяйства и утверждается республиканскими органами народного образования. Для качественного проведения экспертизы УМС НИТ при необходимости к работе РЭК могут привлекаться другие специалисты, которые работают в экспертных рабочих группах, руководимых членом РЭК.

6. РЭК в результате экспертизы УМС НИТ могут выносить следующие решения: рекомендовать или не рекомендовать УМС НИТ к использованию, направлять УМС НИТ в опытную эксплуатацию либо на доработку, рекомендовать УМС НИТ на доводку до уровня готового продукта.

7. Заключение эксперта (рабочей группы) доводится до сведения разработчика УМС НИТ и в случае отрицательного заключения по желанию автора в его присутствии обсуждается на заседании РЭК, после чего РЭК выносит окончательное аргументированное письменное заключение. Отрицательное заключение РЭК не исключает права разработчика УМС НИТ обратиться в другую РЭК.

8. На время экспертизы и после ее завершения РЭК гарантирует авторские права разработчиков УМС НИТ.

9. Труд экспертов (рабочих групп) РЭК подлежит оплате. Оплата труда экспертов (рабочих групп) РЭК осуществляют РЦНИТ, при которых они функционируют, за счет

ассигнований, выделяемых на эти цели соответствующими региональными органами народного образования. Для обеспечения справедливой оплаты труда экспертов (рабочих групп) и исключения произвола в этом вопросе необходима разработка методики определения размера оплаты труда, связанного с экспертизой УМС НИТ.

О подходах к разработке программно-методического обеспечения

Ввиду фактического отсутствия инструментальных средств, специально созданных для разработки ППС, и появления в вузах IBM-совместимой техники представляется возможным использовать для этих целей какие-либо из современных интегрированных систем или систем управления базами данных, развивающихся в направлении интегрированных систем. В качестве таковых могут быть выбраны СУБД dBASE 4, Fox Base 2.10, RBASE, Paradox, DRACLE и др.

Система dBASE 4, по нашему мнению, наиболее удобна для разработки ППС. Во-первых, в режиме диалога она позволяет довольно эффективно создавать многоуровневые pop-up меню, проектировать многостраничные формы ввода, упаковывать изображение экрана в переменные и т. п., что весьма полезно при разработке ППС любого типа. Управляющий язык системы dBASE 4 является простым, структурным и органично встроенным в систему.

Так как dBASE 4 не только представляет интерес для разработки ППС, но и является одной из наиболее интересных разработок по реляционным базам данных, считаем целесообразным:

1. Проработать вопрос о возможности приобретения и использования системы dBASE 4 для учебных целей (хотя бы по одной установке на пединститут; плата в этом случае должна быть чисто символической).
2. Незамедлительно распространить по педагогическим институтам уже приобретенное программное обеспечение, в том числе и то, которое приобретено на ограниченное количество установок.
3. При закупках программного обеспечения для IBM-совместимой техники делать упор прежде всего на инструментальные средства для разработки ППС или на системы, которые могут быть использованы для этих целей.
4. Подготовить к изданию несколько альтернативных пособий по dBASE 4, Fox Base, Framwork и т. п. для студентов пединститутов разных специальностей.
5. Рекомендовать обучение преподавателей институтам на ФПК по системам ти-

па dBASE 4 или Fox Base.

6. Рекомендовать вузам чтение спецкурсов и спецсеминаров по базам данных dBASE, Fox Base или МАСТЕР.

О предмете ОИВТ в средней школе

За последние 20—25 лет складывается новая фундаментальная наука — информатика. В АН СССР в 1985 г. открыто отделение «Информатика, вычислительная техника и автоматизация». Известно, что за последние 10 лет происходят в целом удвоение объемов информации и смена поколений ЭВМ. В развитых странах мира накоплено более 100 млн. персональных ЭВМ, а к 2000 г. планируется полное насыщение общества вычислительной техникой.

В СССР в 1985 г. в условиях дефицита средств ВТ и программного продукта была предпринята попытка прорыва в области информатизации образования в виде:

введения общеобразовательного курса ОИВТ;

создания кафедр информатики в педвузах.

В результате к настоящему времени:

все педвузы оснащены современной ВТ; практически во всех районах страны имеются средние школы, оснащенные средствами ВТ;

разработано много отечественного программно-методического обеспечения;

подготовлено поколение учителей информатики, способных внедрять ВТ в учебный процесс;

подготовлены четыре варианта учебных пособий по дисциплине ОИВТ для школы; начаты работы по реализации следующего этапа информатизации образования в СССР.

Новый этап, очевидно, будет связан с переходом, начиная с педагогических вузов, на новое поколение ЭВМ, программно и аппаратно совместимых друг с другом и позволяющих создавать вычислительные сети, а также базы данных, базы знаний, что позволит перейти на новую информационную технологию обучения.

Все это является предпосылкой реализации полного курса информатики в средних школах и педагогических вузах.

Содержание такого курса должно включать в себя три составляющие: начальную школу, среднюю школу, высшую (педагогическую школу).

В начальной школе предлагается начальное знакомство с ЭВМ и началами информатики.

В средней школе — введение в информатику, элементы информационной технологии. Специализация в средней школе — начала

программирования, базы данных, базы знаний, технология программирования.

В высшей школе — основы информатики, компьютерная технология обучения.

Существующая тенденция ликвидации информатики как самостоятельной учебной дисциплины противоречит развитию науки, техники и общества в целом. Наличие самостоятельного учебного предмета позволит следующее.

1. Образовать центры новых информационных технологий в одном из самых массовых государственных объектов — общеобразовательной школе.

1.1. Внедрить информационную технологию во все школьные дисциплины, предметы.

1.2. Обеспечить эффективное использование ВТ в повседневной жизни школы для разнообразных целей.

1.3. Развить психолого-дидактические положения и принципы применения нового инструмента познания и обучения.

2. Активизировать работу и привлечь внимание руководства школы к проблемам информатизации образования.

3. Обеспечить комфортное развитие и существование обучаемых в XXI в.

4. Избежать парадоксальных ситуаций, возникших в развитых странах мира, где при отсутствии дефицита на программные средства и ВТ качество обучения повысилось незначительно.

Из истории преподавания известен ряд примеров слияния близких предметов, приведших к отрицательным последствиям (алгебра и теория чисел, алгебра и геометрия и др.).

В связи с вышеизложенным, секция считает необходимым рекомендовать Госкомитету по народному образованию:

1. Изменить приказ № 751 в части слияния дисциплины ОИВТ с другими в СШ и базисный учебный план 1990 г. в этой же части.

2. Подготовить на альтернативной основе новую программу по дисциплине ОИВТ.

3. Выделить дополнительные средства на развитие работ, связанных с внедрением новых информационных технологий во все дисциплины школы и педвуза.

О структуре и содержании подготовки педкадров в условиях информатизации образования

Секция считает, что идеология информатизации должна пронизывать подготовку всех категорий педагогических кадров, при этом в одних случаях информатика изучается как предмет со своими методами и пред-

ставлениями, а в других — как средство, инструментарий в области организации учебного процесса. Обсудив на заседаниях вопросы, связанные со структурой и содержанием учебных планов по физико-математическим специальностям, секция отмечает необходимость проведения единой политики подготовки педагогических кадров в области информатики.

В связи с этим секция считает целесообразным рекомендовать Госкомитету СССР по народному образованию, учебно-методическим объединениям по педагогическим специальностям, республиканским министерствам народного образования, ученым советам педагогических вузов при разработке и корректировке учебных планов физико-математических факультетов руководствоваться следующим.

1. В учебных планах по математическим специальностям целесообразно иметь два интегрированных курса: «Информатика» и «Методика преподавания математики и информатики».

Содержание первого из них должно «поглотить» содержание традиционных курсов «Техника вычислений и алгоритмизация», «Основы информатики и вычислительной техники», «Математическая логика и теория алгоритмов», а также содержать элементы дискретной математики (теория графов, теория кодирования и др.), уравнивая доли непрерывного и дискретного в содержании подготовки педагогических кадров.

Содержание второго курса должно включать содержание курсов «Методика преподавания математики», «Методика преподавания информатики», «Использование ЭВМ в учебном процессе».

2. Организовать сквозное изучение курса «Информатика» и последующих дисциплин этого цикла на протяжении всего срока обучения (с 1-го по 10-й семестр). Считать целесообразным организовать изучение курса «Методика преподавания математики и информатики» в интервале с 6-го по 9-й семестр.

3. Объем изучения каждой из указанных дисциплин не должен быть менее 15 % от общего объема времени, отводимого на специальную подготовку.

4. Дальнейшее углубление содержания дисциплин «Информатика» и «Методика преподавания математики и информатики» должно обеспечиваться системой спецкурсов и спецсеминаров.

Секция считает, что создание кафедр информатики и вычислительной техники (или центров информатизации на правах кафедр) целесообразно во всех педвузах страны.

В настоящее время секция считает необходимым расширить работу по экспериментальным учебным планам подготовки учителей информатики, проводимую в Киевском, Свердловском, Славянском пединститутах. Применительно к подготовке учителей всех специальностей необходимо продолжить опытную работу на базе Омского педагогического института.

Статус и направления работы секции

Секция, являясь составной частью ОНМС по информатизации образования, выполняет функции, перечисленные в Положении об ОНМС, концентрируя свою деятельность на вопросах подготовки и повышения квалификации педагогических кадров в педучилищах, педтехникумах, пединститутах, на педагогических отделениях университетов, в институтах усовершенствования учителей, т. е. во всех учебных заведениях и иных подразделениях, осуществляющих подготовку, переподготовку и повышение квалификации педкадров. Секция вырабатывает и выносит на пленум ОНМС свои предложения, рекомендации и другие документы по этому и другим направлениям деятельности ОНМС.

Первоочередными задачами секции являются:

выработка рекомендаций Гособразованию СССР в деле подготовки и повышения квалификации педагогических кадров в области информатизации образования;

анализ и оценка состояния подготовки и повышения квалификации педагогических кадров в педтехникумах, педучилищах, педвузах, выявление тенденций и выработка рекомендаций Гособразованию СССР;

изучение передового опыта в области подготовки и повышения квалификации педагогических кадров, разработки НИТ, обобщение и выработка рекомендаций по их использованию.

С учетом этих задач статус секции оп-

ределяется следующими позициями.

Секция определяет перспективный план издательской деятельности Гособразования СССР и дает рекомендации по составу авторских коллективов для написания учебников, учебных пособий, методических рекомендаций и других изданий.

Учебные планы педучилищ, педтехникумов и педвузов утверждаются Гособразованием СССР только после представления их на рассмотрение секции и получения соответствующей рекомендации.

Секция определяет перспективные направления научных исследований и ОКР по НИТ в образовании и вырабатывает рекомендации Гособразованию СССР по объему их финансирования с обязательным отчетом руководителей тем на секции. Гособразование СССР координирует свою деятельность с АПН СССР, выступая в роли заказчика.

Учебно-методические объединения (УМО) строят свою работу по созданию организационного и учебно-методического обеспечения подготовки педкадров с учетом рекомендаций секции и подотчетны секции в рамках ее компетенции.

Секция считает преждевременной централизацию экспертной работы в области НИТ. Экспертиза должна ограничиваться региональными центрами НИТ, в которых сосредотачивается фонд алгоритмов и программ региона с учетом материально-технического обеспечения региона.

Решения и рекомендации секции должны оперативно доводиться до сведения педагогической общественности и других заинтересованных лиц.

Материал подготовили: М. ЛАПЧИК, председатель секции, канд. пед. наук, проректор по учебной работе Омского педагогического института

В. ЕФИМОВ, ученый секретарь, ведущий инспектор Главной государственной инспекции учебных заведений Гособразования СССР

Ю. ГОРВИЦ, ЦНИИ «Электроника»,
Москва

Развивающие игровые программы для дошкольников

За последние годы нами накоплен определенный опыт в создании программных средств для такой специфической категории пользователей, как дети дошкольного возраста. Пытаясь обобщить и проанализировать этот опыт, мы неизменно приходим к выводу, что наиболее ценным было то, что с самого начала к разработке программ для дошкольников мы подошли как к комплексной проблеме, носящей междисциплинарный характер. Поэтому в создании программ помимо программистов участвуют специалисты разных профессий — психологи, педагоги, эргономисты, художники, музыканты.

Основу программы рассматриваемого класса составляет чаще всего некая психологическая идея, которая, будучи облеченной в «компьютерную» форму, превращается в небольшую управляемую малышом игру, способствующую развитию у ребенка тех или иных способностей.

Но что можно придумать, чтобы малыш 5—6 лет (а может, и младше...), не умеющий ни читать, ни считать, с трудом понимающий взрослую речь, начал «общаться» с компьютером?

Сегодня, разработав уже около сотни программ для детей дошкольного возраста, мы можем не только ответить на этот вопрос, но и поделиться в этой статье своим опытом.

В первую очередь статья адресуется педагогам, психологам и другим спе-

циалистам, не являющимся знатоками компьютерной техники, но желающим воплотить свои профессиональные идеи в виде программ.

Итак, при разработке развивающих игровых программ для дошкольников мы исходим из фундаментальной посылки, что компьютер представляет собой средство воспитания и развития творческих способностей ребенка, формирования его личности.

Необходимо подчеркнуть отличие развивающих компьютерных программ от так называемых коммерческих компьютерных игр. И дело здесь не только в том, что последние, как правило, связаны с насилием, погонями, военным противостоянием, поощрением разрушительных действий, агрессий, что безусловно является недопустимым для дошкольников и младших школьников. Смысл отличия лежит гораздо глубже и состоит в том, что для «коммерческих» игр характерен жесткий режим взаимодействия с играющим, при котором игрок должен «отвечать компьютеру», следуя условиям, возникающим на экране, и в темпе, чаще всего диктуемом компьютером. При этом дети сравнительно быстро овладевают навыками управления устройством ввода (клавиатура, различные манипуляторы) и их внимание целиком поглощается содержанием программы, захватывающей своим сюжетом. «На этой стадии интеллектуальная активность ребенка резко снижается,

так как инициатива в игре принадлежит не ему, а компьютеру. Таким образом, «коммерческие» компьютерные игры навязывают игроку диалог, в ходе которого происходит как бы «персонафикация» компьютера — компьютер воспринимается как партнер по игре, а не как средство игры» (С. Л. Новоселова).

Зададимся вопросом: что отличает компьютер от любого другого игрового средства? Какими особыми свойствами обладает компьютер по сравнению с обычной игрушкой или иным техническим устройством? И наконец, какие особенности компьютера следует учитывать при использовании его как нового средства деятельности дошкольников?

Опыт разработки развивающих игровых программ, составляющих комплект компьютерных игр «Знайка», анализ отдельных программ системы «Рига» и некоторых учебных программ зарубежных фирм позволяют ответить на эти вопросы.

Прежде всего, компьютер представляет собой «интерактивное» средство. Это означает, что ребенок может активно управлять игрой, опосредованно (через клавишные или иные средства ввода информации) воздействуя на экранный видеоряд и другие компьютерные средства. Здесь эффективно используется возможность программирования клавиатуры — назначения произвольных функций отдельным кнопкам клавиатуры. Обработка нажатий на отдельные клавиши или блокирование нажатий осуществляются программным путем, непосредственно с помощью средств игровой программы. Так появляется возможность управлять сюжетом игры путем нажатия тех или иных клавиш, контролировать при этом автоматически правильность нажатия.

Компьютер может воспроизводить текстовую, графическую, цветовую, звуковую информацию, управлять с помощью выводимых на специальный разъем (порт ввода-вывода) электрических сигналов различными устройствами и приборами, а также принимать информацию через этот порт. Следовательно, компьютер можно рассматривать как универсальную игрушку, меняющую свое

назначение (функциональное назначение, смысл) при смене программы.

Компьютер рассматривается как дополнительное педагогическое средство в системе средств развития ребенка. Из этого следует, что наиболее эффективными являются компьютерные игры, которые непосредственно соотносятся с играми некомпьютерными или позволяют включить компьютерную игру в сюжетно-ролевые и другие организационные формы игры детей дошкольного возраста.

В компьютере имеется возможность генерации случайного числа с помощью программно или аппаратно реализованного датчика. Это позволяет формировать вариативные игры, в которых резко возрастает количество и разнообразие дидактического материала. Эту особенность следует учитывать при составлении «обучающих» компьютерных программ. В них подготовка дидактического материала ведется компьютером либо случайным образом, либо по заранее предусмотренному алгоритму (на самом деле, естественно, элементы материала закладываются программистом в содружестве с педагогом, психологом и другими специалистами). Такой подход позволяет воспитателям и методистам детского сада освободиться от рутинной работы по подготовке материала для занятий с детьми. Закрепление навыков и умений при работе с этими программами происходит в игровой компьютерной форме.

Важным свойством компьютера является возможность накопления в ходе компьютерной игры введенной ребенком информации и ее программного анализа для объективного учета индивидуальных особенностей ребенка. Анализировать можно и введенный ответ, и серию ответов. Тем самым программу можно «подстраивать» под конкретного ребенка, предъявляя ему задания в удобном для него темпе и соответствующем его уровню развития качестве.

Игровые программы, в которых осуществляется моделирование тех или иных процессов, событий, позволяют малышам самим управлять компьютерными моделями реальных длительных процессов и наблюдать на экране результа-

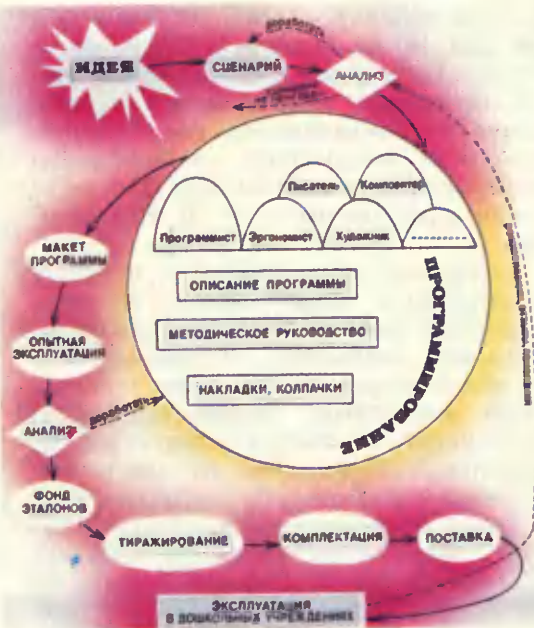
ты и последствия этих процессов.

Огромные возможности открываются, если к компьютеру присоединить через порт ввода-вывода разнообразные периферийные устройства и начать программно ими управлять. Это резко расширяет области применения компьютера в дошкольных учреждениях. Мы, в частности, исследуем и отрабатываем приемы и средства управления с помощью компьютера электромеханической игрушкой; ввода информации в компьютер с помощью неклавиатурных устройств ввода; анализа информации, поступающей от различного рода датчиков; управления нагрузочными устройствами в физкультурных тренажерах и приспособлениях, способствующих физическому развитию детей; управления диагностическим и лечебным медицинским оборудованием в детских лечебных учреждениях и т. д.

Благодаря соединению компьютеров в локальную информационно-вычислительную сеть появляется возможность компьютерной реализации диагностических методик. Для детей дошкольного возраста их можно оформить в виде игровых программ. В процессе игры в памяти компьютера формируется банк данных, который затем пересылается по сети в узловую (центральную в случае простой звездообразной топологии сети) машину для последующей обработки средствами пакета программ математической статистики. Важно, чтобы реализации на компьютере подвергались апробированные методики, созданные специалистами в соответствующей прикладной области.

Остановимся на особенностях технологии разработки детских программ. Как показывает опыт, наиболее удачными являются те, в которых учитываются закономерности мышления и восприятия ребенка, особенности игровой деятельности дошкольников, уровень их умственного развития, эргономические параметры, в том числе антропометрические, функциональные и др. Все это диктует специфическую технологическую цепочку разработки, которую можно рекомендовать на основании накопленного опыта (рис.).

На основании сформированной (но



постоянно обновляемой) концепции использования компьютерной техники в дошкольном воспитании, исходя из задач разностороннего развития детей дошкольного возраста формулируются идеи развивающих, обучающих, диагностических и других компьютерных игр. Автор идеи разрабатывает ее, доводя до уровня сценария. Это достаточно слабо формализованное понятие, но опыт позволяет утверждать, что сценарий компьютерной программы должен включать следующие разделы: 1) название игры; 2) сведения об авторе (соавторах) и организации-разработчике; 3) год разработки; 4) назначение; 5) описание.

Так как четыре пункта не нуждаются в пояснениях, остановимся на последнем. Описание может представлять собой словесное или схематичное представление сюжета программы, используемых графических образцов видеоряда, задействованных и запрещенных клавиш, последовательности действий ребенка и ответных изменений на экране дисплея. Причем важно, чтобы на «дереве» состояний и переходов программы (математически любой алгоритм программы можно представить в виде графа) не было тупиковых ветвей — в программе не должно быть ситуации, в кото-

рой она «не знает, что делать дальше». Описывать игру следует как можно подробнее, вплоть до таких мелочей, как, например, в каком месте экрана должен появляться тот или иной образ, какого он должен быть цвета и т. п. Все эти детали могут иметь принципиальное значение для успешной игры ребенка.

Такие требования к описанию обусловлены разделением труда — сценарий пишет один человек (люди), а программу — другой (другие). От качества сценария во многом зависят качество будущей программы, срок и эффективность ее реализации. Практика показывает, что принцип, согласно которому постановщику задачи (сценаристу) следует предоставить удобный инструмент для реализации идеи, на практике неприемлем.

Итак, составленный сценарий должен пройти психолого-педагогическую экспертизу на соответствие целям развития и воспитания и программно-техническую — на предмет реализуемости в условиях использования данного компьютера или инструментального программного средства (языка программирования). При положительном результате экспертизы программист приступает к реализации сценария. Следует отметить, что квалификация программиста во многом определяет не только качество будущей программы, но иногда и принципиальную возможность рождения программы как таковой.

Известны случаи (из практики автора), когда на обдумывание способа переноса уже готовой программы в более благоприятную на первый взгляд программную среду (перевод программы с языка Фокал на язык Бейсик) и на изобретение специального механизма хранения и извлечения информации уходило несколько месяцев.

На этом этапе создается еще не программа, а макет. При этом крайне нежелательно максимально тесное сотрудничество автора сценария и программиста, позволяющее в процессе разработки уточнять необходимые детали. Программа подвергается испытаниям. Не буду останавливаться на методах испытаний, укажу лишь, что основная

их цель — выявить и исправить наиболее очевидные ошибки, довести программу до уровня, приемлемого для передачи программы в опытную эксплуатацию. Этот этап проходит в дошкольном учреждении, где опытные педагоги, уже владеющие методиками применения компьютеров, в режиме реальной игры ребенка за компьютером выявляют все особенности использования данной программы, а иногда и ее частичную или даже полную непригодность. В этом случае программа возвращается на доработку, подвергаются корректировке и сценарий и программа. Этот процесс носит итеративный характер, и его следует считать вполне закономерным.

Параллельно с разработкой программы ведется разработка описания, методического руководства и вспомогательных материалов (клавиатурных накладок, колпачков и т. п.). Этим документам необходимо уделять особое внимание, ибо их наличие обуславливает высокое качество программного продукта как изделия.

Отечественный и зарубежный опыт говорит о том, что зачастую именно отсутствие качественной документации (например, в случае несанкционированного копирования программного средства) приводит к неэффективному использованию либо вообще к отказу от применения программы.

Подготовка описания, в котором излагаются назначение игры и ее правила, действия ребенка и ответные реакции программы, методических руководств, в которых предлагаются методические приемы использования программы с педагогической точки зрения, а также всех сопутствующих материалов ведется в аналогичном итеративном режиме.

После приемки готовой программы со всеми упомянутыми документами изделие подлежит тиражированию для передачи в дошкольные учреждения. Интересно отметить, что на этом разработка программы не должна заканчиваться! Необходимо учитывать обратную связь, возникающую между практическими работниками дошкольных учреждений, эксплуатирующими конкретную

программу, и ее разработчиками. Очевидно, что практика может выявить незамеченные ошибки или недостатки в программе и «скрытые резервы», в частности новые методические приемы использования.

Отдельно следует остановиться на эргономическом проектировании программ для дошкольников. Это связано с необходимостью обеспечить для ребенка функциональный комфорт (Л. Д. Чайнова).

Как показали предварительные исследования (они еще продолжаются), интегральный показатель функционального комфорта для детей дошкольного возраста необходимо оценивать, исходя как минимум из трех компонентов: мотивации — дети должны играть заинтересованно, охотно; преодоления в игре посильных трудностей смыслового характера и получения удовлетворения от их преодоления; минимизации затрат на преодоление технических трудностей.

Экранные образы, способы их трансформации, изображения условных символов на клавиатурных накладках (колпачках) и их расположение, способы взаимодействия ребенка с компьютером, возможные ответные сигналы компьютера на действия ребенка и другие возможности игровой программы — все это объекты для эргономического проектирования.

1. **Экранные образы.** Так как дети дошкольного возраста не умеют читать либо только учатся, текстовая информация в программах для дошкольников практически отсутствует (не считая заставки-титала программы). Исключения составляют программы по развитию речи и письма, где используются буквы увеличенного размера. Побочным положительным эффектом такого подхода является то, что такие программы можно считать независимыми от национального языка и эффективно использовать в любых национальных дошкольных учреждениях и системах образования. В тех случаях, когда программа должна вывести ребенка вопрос или дать задание, применяются условные знаки типа пиктограмм, понятные для ребенка. Следует избегать применения стандартного шрифта, заложенного в знако-

генераторе компьютера, как не обеспечивающего необходимый для детей размер букв, цифр и других знаков.

Необходимо тщательно продумывать цветовое решение изображения на экране. Учитывая ограниченность цветовой гаммы и, как показал анализ, негативное влияние черного фона экрана на восприятие, а также раздражающее воздействие красного цвета, желательно стремиться к использованию красного цвета только для указания аварийной ситуации, в качестве сигнала ошибки или запрещения действия, а также в качестве вспомогательного (не основного) цвета во многокомпонентных изображениях; использовать зеленый цвет как поощряющий или разрешающий действие; ограничивать количество черных объектов на экране, главным образом это относится к фону.

Хотя, с одной стороны, психологам рекомендуется делать сюжет игры понятным ребенку, для чего зачастую игру стараются оформить в виде сказочного сюжета со знакомыми ребенку персонажами, с другой стороны, рекомендуется не «засорять» экранное изображение множеством мелких «псевдореальных» деталей, поскольку они отвлекают ребенка от решаемой задачи, снижая тем самым развивающий эффект игры.

Предпочтительнее для такого возраста абстрактные, многозначные объекты, позволяющие ребенку самому наделять их смыслом, фантазировать. В этом важная роль отводится программам, в которых конкретная цель отсутствует, ребенок формирует цели самостоятельно по ходу игры (так называемый класс «режиссерских игр»). Изображения на экране должны быть достаточно крупными, их взаимное расположение обычно диктуется условиями игры.

2. **Способы трансформации образов.** К способам трансформации относятся способы перемещения образов по экрану, удаления (стирания) с экрана, преобразования одних образов в другие, перекрашивания, изменения размеров, ориентации, взаимного расположения образов. Важно отметить, что для маленьких детей обязательно следует учитывать темп преобразований. Он не должен быть слишком быстрым, поскольку скорость



восприятия образа, оценки ситуации и реакции на нее у детей в значительной степени отличается от взрослых.

3. Изображения на клавиатурных накладках. Если для детей школьного возраста считается предпочтительным использовать традиционную компьютерную клавиатуру как один из элементов подготовки к будущей профессиональной деятельности, то для детей дошкольного возраста она избыточна. Поэтому для компьютеров с пленочной (низкопрофильной) клавиатурой высокоэффективными оказались клавиатурные наклейки, на которых условными знаками выделяются только те клавиши, которые непосредственно используются в данной игре. Действие остальных клавиш при этом рекомендуется блокировать программным путем. Одним из важнейших эргономических требований является расположение основных клавиш для манипуляций (в частности, клавиш перемещения курсора) под правой рукой (для правшей). Кроме того, во многих играх цифровые клавиши расположены так, как это принято в устройствах цифрового набора (кассовых и телефонных аппаратах, калькуляторах и т. п.) в случае, если на компьютере отсутствует специальная цифровая секция клавиатуры. вспомо-

гательные клавиши, в особенности те, что предназначены для педагогов или экспериментаторов, должны отличаться по цвету, расположению или иным способом от функциональных игровых клавиш.

Удобно какую-либо одну клавишу (например, клавишу «пробел») выделить для осуществления функции перехода к началу игры (неаварийного завершения программы).

4. Способы взаимодействия ребенка с компьютером. Важнейшим показателем является простота интерфейса, благодаря которой как раз и достигается понимание ребенком способа действия для перехода к следующей стадии — собственно игровой деятельности. Программно интерфейс реализуется чаще всего в виде пиктографического меню, где манипуляциями ребенок выбирает один из предложенных вариантов ведения игры. Другой рекомендуемый способ — непосредственное взаимодействие, когда при нажатии на определенные клавиши немедленно отрабатываются соответствующие действия.

5. Ответные сигналы компьютера. В тех играх, где требуется бинарная реакция (типа «правильно — неправильно»), возможны различные способы демонстрации этой реакции: графическим символом, носящим условный характер (зеленый круг — хорошо, красный — плохо), осмысленным графическим образом, а потому представляющим «ценность» для ребенка (улыбающееся либо огорченное личико), звуковым эффектом (ободряющая, веселая мелодия — грустная мелодия).

Аналогичные цветковые, звуковые, комбинированные сигнальные эффекты можно применять для фиксации внимания ребенка в каких-либо определенных ситуациях. Полезно отметить, что при индикации негативного результата не следует использовать «разрушающих» или «агрессивных» компьютерных эффектов, провоцирующих детей на повторение такой ситуации ради самого эффекта.

В заключение следует отметить некоторые особенности модели компьютера «Электроника БК», для которой разработан комплект программ «Знайка».

Некоторые из этих особенностей относятся и к другим моделям ПЭВМ аналогичного класса.

Главным недостатком следует считать ограничение по памяти, существенно влияющее на содержание программ. Размер свободной области для пользователя программ составляет 16К байт, что для программ на языке Бейсик (с учетом особенностей реализации самого интерпретатора языка) означает порядка 300 строк на программу. Из-за этого реализуются лишь программы, ограниченные по функциям, или по изобразительным возможностям, или по логике.

Отсутствие таймера (аппаратно реализованного) не позволяет эффективно использовать действия, связанные с отсчетом времени. Работу таймера приходится имитировать программным путем, что связано с определенными программными ухищрениями.

Ограничения цветовой гаммы и разрешающей способности экрана не позволяют формировать изображения очень высокого качества. Поэтому приходится пользоваться специальными методами программирования.

Отсутствие встроенного накопителя на магнитном диске вынуждает всю необходимую информацию хранить непосредственно в оперативной памяти компьютера, а с учетом сказанного выше это существенно сужает возможности программ.

Соединение компьютеров в сеть типа «звезда», что ранее отмечалось как достоинство компьютерного комплекса, выступает как ограничение — компьютеры не связаны между собой (а лишь с центральной машиной). Из-за этого отсутствует возможность обмена информацией между компьютерами, а с точки зрения методики их использования отсутствует возможность организации целого направления игр — коллективных игр для дошкольников, представляющих особую важность и педагогическую ценность.

Не хватает также встроенных аппаратных и соответственно программных средств формирования динамических изображений (спрайтов, имеющих в

некоторых компьютерах), что существенно ограничивает возможности создания движущихся объектов размером больше стандартного курсора. Для компенсации указанного недостатка приходится прибегать к искусственным приемам программирования вплоть до разработки специальных инструментальных программных средств, таких, как редактор спрайтов и спрайт-программа. Следует отметить, что реализация динамических объектов таким способом связана с рядом технических трудностей и ограничений.

Помимо указанных методов применяются такие приемы, как отработка кодов сдвижки/раздвижки строки, перерисовка объектов путем их быстрого стирания и вывода в новом месте экрана. Последний способ неэффективен в случае крупных разноцветных образов. Еще одним методом являются запоминание в памяти отдельных фрагментов памяти видеОЗУ (специальной области памяти, где хранится образ экрана в некотором закодированном виде) и обмен данными между этими областями оперативной памяти. Реализация этого способа требует точности выполнения, а также знания технических тонкостей устройства данной конкретной модели ЭВМ. Применение этого приема приводит к потере программной (пусть и относительной) переносимости на другие модели компьютеров.

Поскольку одним из эффективных средств воздействия на человека — пользователя ЭВМ, в данном случае ребенка, является звук, то принципиальным недостатком следует считать отсутствие синтезатора звука (хотя бы простейшего), не говоря уже о синтезаторе речи. Тем не менее с помощью определенных средств и методов программирования можно не только извлекать звук в достаточно широком диапазоне, но даже формировать речь. Отдельно следует рассматривать возможность подключения синтезатора звука и речи в качестве дополнительных периферийных устройств. В необходимых случаях рекомендуется использовать звуковое сопровождение программ для дошкольников.

МОЛОДЕЖНАЯ ИНИЦИАТИВА

В номере:

- Новые разработки студентов и школьников
- Рецензия на новую книгу издательства «Мир»
- Организована студенческая секция ВОИВТ

«ДЕМО» — язык для начинающих

В прошлом выпуске «М. И.» была опубликована статья, в которой содержалась краткая информация о созданной в МКБ при ИПИ АН СССР Объединенной библиотеке языка Си для компьютеров класса MSX. Там отмечалось, что на ее базе был написан ряд самостоятельных прикладных программ. Одной из этих работ является интерпретатор языка «ДЕМО». Этот язык разработан в МКБ и служит для описания демонстрационных и контрольных задач.

Название языка — «ДЕМО» — это сокращение от английских слов «DEMOstrative» — наглядный, «DEMOstration» — демонстрация (наглядными примерами). Название отражает основную цель, которая преследовалась при разработке языка, — предоставить любому, даже очень слабо знакомому с информатикой, чело-

веку некий аппарат, с помощью которого он мог бы максимально быстро и просто привлечь компьютер к своей работе: возложить на него часть функций рекламного агента или социолога, действующей демонстрационной модели или контролера знаний. При этом основной расчет делался именно на школьных преподавателей, которые рады бы использовать имеющуюся в школе технику, но, к сожалению, не знают как.

Поэтому и родилась идея создания простого алгоритмического языка, который бы помог педагогам самостоятельно освоить азы программирования и предоставить возможность использовать компьютеры в преподавании различных предметов.

Одна из проблем, с которой сталкивается человек при первом знакомстве с программированием, — это не-

обходимость знания английского языка и определенных, часто жестких и не очень простых, правил языка программирования. В языке «ДЕМО» эти проблемы частично или полностью решены.

Во-первых, все операторы языка — привычные нам русские слова. Во-вторых, все арифметические действия реализованы с помощью операторов, а не с помощью математических символьных записей. И в-третьих, правила написания программ и синтаксис операторов реализованы так, чтобы текст программы наиболее походил на естественное, минимально формализованное, изложение хода (алгоритма) работы.

Статья не преследует цель научить читателя писать на языке «ДЕМО». Прежде всего хотелось бы привлечь внимание заинтересованных

людей к сфере школьной информатики и тем самым помочь ее развитию. Поэтому дается максимально краткий обзор возможностей языка и связанных с ними направлений его применения.

Возможности языка «DEMO»

Все операции можно подразделить на пять типов:

1. Работа с готовыми файлами:

ЗАГРУЗИ — загрузить бинарный файл в память;

ЗАПУСТИ — запустить бинарный файл в памяти;

108 **НАЭКРАН** — загрузить на экран картинку;

ЗАПОМНИ — записать картинку с экрана в файл;

ЛИСТАЙ — постранично распечатать на экран текстовый файл;

ДАЛЕЕ — закончить работу путем запуска другой программы.

2. Работа с экраном:

ТЕКСТ — установить один из текстовых режимов работы экрана;

ГРАФИКА — установить один из графических режимов работы экрана;

КУРСОР — установить курсор (с выводом или без) в нужное место экрана;

ПЕЧАТАЙ — напечатать текст на экран в любом режиме работы;

ЦВЕТ — установить цвет изображения, фона и бордюра;

СОТРИ — очистить экран;

ТОЧКА — нарисовать точку;

КРУГ — нарисовать окружность;

СОЕДИНИ — нарисовать линию, незакрашенный и закрашенный прямоугольники;

ЗАКРАСЬ — закрасить замкнутую область.

3. Опрос клавиатуры:

СПРОСИ — ввод строки;

ЖДИ вр. — ожидать ввод с клавиатуры нужное время;

ЖДИ — ожидать целевой ввод с клавиатуры;

ПРОБЕЛ — выполнение набора действий по нажатию клавиши «пробел».

4. Работа с переменными:

ПОЛОЖИ — присвоить значение переменной;

ПРИСВОЙ — присвоить переменной адрес строки для косвенного вывода;

ВЫВЕДИ — прямой и косвенный вывод переменных и строк;

СРАВНИ — операция арифметического сравнения;

ПРИБАВЬ — операция сложения;

ОТНИМИ — операция вычитания;

УМНОЖЬ — операция умножения;

РАЗДЕЛИ — операция деления.

5. Изменение работы программы:

ЕСЛИ

ЕСЛИНЕ — осуществить условное ветвление программы;

ИДИ — осуществить безусловное ветвление программы;

ИМЯ — осуществить вызов подпрограммы;

ВОЗВРАТ — вернуться из подпрограммы;

ПОВТОРИ — повторить набор действий (цикл);

КОНЕЦ — закончить работу программы путем выхода в MSX.

Предложения, заключенные в скобки, считаются комментариями и игнорируются

интерпретатором. Ниже, в примерах, они будут использованы для пояснения программ.

Что же позволяет делать «DEMO»?

Сочетая текстовую информацию с наглядной графической, учитель может проводить уроки в виде лекций, которые «читает» компьютер. При этом каждый слушатель может заниматься в необходимом темпе и имеет возможность возвратиться к любой части лекции. Переменяя лекцию вопросам, преподаватель получает возможность одновременного контроля степени понимания нового и усвоения старого материала. Таким образом преподаватель создает не что иное, как систему для изучения предмета методом погружения по своей методике.

Указывая место ввода информации в различных местах текста на экране, преподаватель языка может задавать упражнения для проверки или закрепления знаний орфографии и(или) пунктуации.

Быстрая графика позволяет выполнять анимацию рассчитанных математических моделей.

Имея возможность отражать результаты текстовых ответов на вопросы в чистлах, преподаватели могут оперативно проводить тестирование и анкетирование и, кроме того, сразу получать обработанные результаты.

Язык «DEMO» дает возможность работать с готовыми данными и программами. Человек может использовать написанные программистами вставки, тем самым расширяя возможности языка в необходимом направлении.

Примеры, приведенные ниже, демонстрируют различные сферы применения языка «DEMO».

Пример контроля знаний по географии:

:Начало
Спроси Столица США;
Если Вашингтон [Печатай Правильно; Конец;]
Печатай Неверно! Столица США — Вашингтон.;

Пример контроля знаний по орфографии русского языка:

:Начало
Печатай Поставьте правильное окончание;
:1
Печатай
У Маши нет красн??? шарфа;
Прибавь $\#1,2$; (Прибавляю два к переменной $\#1$)
Курсор 17, $\#1$; (Установка места ввода по переменной)
Спроси;
Если Не ого [Печатай Неверно!; Иди 1;]
Печатай Правильно;

Пример контроля знаний по иностранному языку:

:Начало
Печатай Переведите на английский язык слова;
Спроси стол→; (Везде в случае верного ответа)
Если table Верно; (вызывается подпрограмма Верно)
Спроси карандаш
Если pencil [Верно;]
Спроси окно→;
Если window [Верно;]
Спроси стена→;
Если wall [Верно;]
Спроси человек→;
Если man [Верно;]
Печатай Ваша оценка;
Выведи $\#1$; (В переменной 1 содержится оценка)
Конец;
:Верно (Подпрограмма подсчета оценки)
Прибавь $\#1,1$; (в случае правильного ответа)
Возврат;

Пример для демонстрации графических возможностей:

:Начало
Цвет 15, 1, 1;
Графика 2; (Установка графического режима работы)
Положи $\#1, 81$;
:1
Повтори 3, 22 (Линия движется слева направо)
Соедини 125, 50, $\#1, 100, 1, Л$;
Прибавь $\#1, 4$;
Соедини 125, 50, $\#1, 100, 15, Л$;
:3
Пробел [Конец;]
Повтори 4, 22 (Линия движется справа налево)
Соедини 125, 50, $\#1, 100, 1, Л$;
Отними $\#1, 4$;
Соедини 125, 50, $\#1, 100, 15, Л$;
:4
Пробел [Конец;]
Иди 1;

Надеемся, что заинтересовать вас этой областью применения школьной вычислительной техники удалось. Все ваши предложения и вопросы ждем по телефону в Москве 132-19-33.

Кирилл ЧУРЮКАНОВ

Добро пожаловать в мир компьютера

Недавно выпущенная издательством «Мир» книга «Основы компьютерной грамотности»* стала логическим продолжением серии переводной литературы этого издательства, посвященной научно-популярному изложению ключевых вопросов «computer sciences» (информатики).

В последнее время лавинообразно растет поток различной литературы, посвященной основам информатики и адресованной различным группам учащихся. Однако простое вливание данных и фактов в головы учеников почти ничего не дает и никак не влияет на качество подготовки слушателей различных учебных заведений. Необходимо творческое осмысление материала, его проработка на конкретных задачах из реальной жизни. Только интерес ученика к проблеме, поставленной перед ним для самостоятельного решения, поможет интенсифицировать процесс обучения и позволит поновому подойти к задаче формирования творческой и всесторонне развитой личности.

* Кершан Б., Новембер А., Стоун Дж. Основы компьютерной грамотности. М.: Мир, 1989.

Книга «Основы компьютерной грамотности» является одной из немногих попыток максимально связать изложение нового материала, в данном случае вопросов информатики, с требованиями повседневной практики. Написанная живым и доступным языком (немаловажная заслуга в этом принадлежит коллективу переводчиков), книга хорошо продумана в методическом плане и является своеобразной стартовой площадкой для всех, кто решил постигнуть «тайны» компьютера.

Книга состоит из четырех частей и четырнадцати глав, каждая из которых кроме изложения основного материала содержит разделы «Опорные точки» и «Вопросы для повторения», где авторы стараются акцентировать внимание читателя на основных идеях главы и предлагают вопросы для самоконтроля. Каждая глава содержит также раздел «Повторение пройденного», в котором еще раз формулируются основные положения изложенного материала, приводится перечень новых терминов вместе с толкованиями и объяснениями, перечисляются вопросы для самостоятельного обсуждения. Все это дает возможность читателям в более выпуклой

форме представить затронутые в главе проблемы и сформировать свое собственное суждение о них.

Предполагается, что читатель параллельно с изучением теоретических положений книги сможет закреплять полученные навыки на компьютере, поэтому в каждой главе предусмотрен раздел «Практические занятия», где объясняется последовательность всех манипуляций, которые необходимо проделать с компьютером (инструкции приведены для компьютеров различных типов: IBM PC, Apple II, Commodore, TRS-80), и список конкретных заданий для выполнения.

Неоценимую помощь преподавателю, использующему книгу в качестве учебного пособия, окажут контрольные вопросы, завершающие каждую главу и построенные либо по принципу выбора правильного ответа на вопрос по теме данной главы (из спектра предложенных), либо требующие от читателя ввода недостающих слов в предложения, излагающие отдельные положения предложенного материала.

Часть 1, «Компьютеры в нашей жизни», знакомит нас с компьютером как инструментом познания окружающей действительности, который помогает вам дома и на работе. Изложение начинается с краткого экскурса в историю развития средств вычислительной техники, объяснения назначения и порядка работы периферийных устройств. Предельно четко и доступно освещаются основные принципы работы компьютера, вводятся понятия бита, байта, операционной системы и прикладной программы.

Часть 2, «Возможности программ», посвящена ответам на вопросы: что такое язык программирования? Ка-

кие существуют конструкции языка? Что такое редактирование программ? Объяснение азов программирования ведется на примере Бейсика — наиболее доступного и простого для неподготовленного читателя языка общения с машиной, которым, однако, не надо чрезмерно увлекаться (следует ясно видеть границы его применимости).

Авторы книги стараются приобщить читателя к культуре (искусству) проектирования программ, знакомят его с различными способами построения программных комплексов, правилами тестирования и документирования.

Часть 3, «Использование прикладного программного обеспечения», целиком посвящена обсуждению вопросов применения компьютеров для решения различных прикладных задач. Читатель получит представление о программах обработки текстов, позволяющих редактировать документы и красиво выводить их на печать, узнает о том, что такое база данных и как с ней работать и, наконец, освоит правила ведения электронных таблиц и научится использовать их в своей повседневной практике.

Важное место в этой части книги уделяется машинной графике. Изучив главу 11, читатель сможет с помощью компьютера рисовать различные картинки, получит представление о том, где и как применяется машинная графика, узнает о различных устройствах графического вывода и о том, как происходит, например, вывод изображения на бумагу.

Часть 4, «Компьютеры сегодня и завтра», рассказывает о том, как с помощью компьютера организовать общение на расстоянии, как использовать компьютер в магазине и в сфере обра-



зования. Последняя глава книги целиком посвящена рассмотрению наиболее перспективных, с точки зрения настоящего дня, приложений компьютера. Это и экспертные системы, и безбумажные технологии, искусственный интеллект, робототехника, распознавание и интерпретация речи.

Как правило, читатель, впервые столкнувшийся с информатикой, испытывает определенные затруднения с терминологией, которые усугубляются недостаточным знанием английского языка. Именно поэтому в конце книги приведен русско-английский словарь-минимум по основным терминам из области «computer sciences».

В целом книга «Основы компьютерной грамотности» представляет собой удачное сочетание научного изложения материала и популярной формы его представления, хорошо подобранных примеров и практических заданий, методически обработанных приемов обучения, учитывающих особенности психологии ученика.

Книга будет безусловно полезна широкому кругу читателей, желающих путем самообразования получить знания по компьютерной тематике. Учителям можно рекомендовать использовать данную книгу при подготовке курсов, связанных с компьютерными дисциплинами.

Д. ВОЛКОВ

кетах, на одной дискете (720К) можно хранить до 100 вариантов контрольных работ.

Достоинством программы является то, что интерфейс пользователя позволяет производить все процедуры без привлечения команд операционной системы, т. е. рассчитан на неквалифицированного (простите, учителя) пользователя.

Программа позволяет проводить одновременный опрос нескольких учащихся, при этом учитель с центральной ЭВМ пересылает задания учащимся на их рабочие места (терминалы), имеет возможность контролировать любого ученика и даже задавать время на проведение опроса. Следует заметить, что при загрузке контрольной работы ЭВМ автоматически формирует для каждого учащегося индивидуальное задание.

Взаимодействие отдельных блоков программы координируется особой программой МЕНЮ. Кроме того, каждый блок программ содержит подробные инструкции пользователю, которые можно получить практически в любой момент работы программы.

Остается добавить, что пакет разработан на ПЭВМ «Ямаха» на Бейсике в ОС MSX v. 2.1. Объем памяти, необходимый для работы пакета, не более 15К.

Программа прошла апробацию в школе по курсу «Химия». В конкурсе прикладных программ на конференции по информатике учащихся Ленинграда (февраль 1990 г.) программа заняла первое место.

Адрес для справок: 197371, Ленинград, Комендантский пр., д. 28, корп. 2, кв. 171. Ионесу Андрею.

**Андрей ИОНЕС,
Юрий АБАЯНЦЕВ**

Нам хотелось помочь...

Мы, ученики XI класса ФМШ № 30 Ленинграда, видим поистине каторжный труд наших учителей. Бесконечные опросы, зачеты, контрольные — и все это надо проверить (а ведь в классах по 30—40 человек)! Наши родители — учителя, и поэтому мы, наверное, лучше других знаем, сколько времени уходит на проверку одной работы (особенно в старших классах, где ее объем подчас достигает 8—10 листов).

Как известно, одна из наиболее распространенных форм работы с учащимися — так называемый программируемый опрос, когда к заданному вопросу прилагается несколько вариантов ответа и требуется найти верный. Такие опросы эффективны тогда, когда надо оценить знания больших групп учеников.

Наблюдая работу учителей, если можно так выра-

зиться, изнутри и снаружи, нам захотелось чем-нибудь облегчить их труд. В течение этого года мы разработали пакет обучающих программ. Он позволяет проводить программированные опросы практически по любым предметам.

Пакет представляет собой интегрированную среду с развитым интерфейсом пользователя. В состав пакета входит программа РЕДАКТОР, позволяющая учителю создавать новые варианты вопросов, редактировать их, производить коррекцию базы данных (т. е. корректировать разработанные контрольные).

Таким образом, учитель имеет возможность создавать, обновлять, корректировать базу данных контрольных опросов практически по любому предмету. Программа позволяет учителю сохранять варианты контрольных опросов на магнитных дис-

Студенческая секция ВОИВТ

112

В марте 1990 г. Центральное правление Всесоюзного общества информатики и вычислительной техники приняло решение об организации студенческой секции ВОИВТ — новой общественной организации, объединяющей на добровольных началах студентов, специализирующихся в области информатики и вычислительной техники; предприятия, организации и частных лиц, заинтересованных в развитии творческого потенциала студенческой молодежи. Главной задачей секции является оказание всесторонней помощи и поддержки талантливой студенческой молодежи. Деятельность студенческой секции регламентируется Положением о секции и Уставом ВОИВТ. Членами секции могут быть преподаватели школ, техникумов и вузов, научные и инженерно-технические работники, студенты высших и средних специальных учебных заведений, учащиеся специализированных и общеобразовательных школ, СПТУ, а также государствен-

ные и совместные предприятия, кооперативы, другие производственные и общественные организации. Базовая организация секции — Московский институт электронного машиностроения.

Основные направления работы секции:

пропаганда научно-технических достижений в области информатики и вычислительной техники, воздействие на формирование общественного мнения по проблемам информатизации общества, издание собственной газеты; профессиональный отбор талантливой студенческой молодежи;

содействие персональному распределению молодых специалистов по заявкам предприятий и организаций; обеспечение студентов работой по специальности в хозрасчетных студенческих центрах научно-технических услуг, финансирование перспективных инициативных разработок;

привлечение студентов к участию в научно-технических конференциях, семинарах, аукционах, выставках,

ярмарках идей, смотрах и конкурсах;

издание и распространение научной, научно-популярной и методической литературы по информатике и вычислительной технике; организация обучения студентов иностранным языкам;

развитие международных связей и взаимовыгодного сотрудничества со студенческими и молодежными организациями, научной и инженерно-технической общественностью зарубежных стран, содействие активизации международного научного туризма.

В соответствии с Уставом ВОИВТ студенческая секция может создавать хозрасчетные студенческие центры научно-технических услуг. Уже утвержден Устав первого такого центра — «Архонт». В ближайшее время ХСЦНТУ «Архонт» приступит к работе.

Ждем ваших предложений. Информацию о работе студенческой секции ВОИВТ и ХСЦНТУ «Архонт» можно получить у секретаря секции Щербининой Ирины Анатольевны по московскому телефону 297-20-52, доб. 2-89.

А. ЛУКЬЯНОВ, Р. САЛИМЖАНОВ, В. СИТНИКОВ

О модели выпускника средней школы по курсу ОИВТ

В статье академика А. Самарского «Компьютер — инструмент познания» (Известия, 1988. 2 августа.) отмечается, что «отставание в компьютерной технике вряд ли преодолеем в ближайшие годы». В связи с этой, разделяемой авторами настоящей статьи, мыслью нам кажется целесообразным обсудить вопросы, связанные с преподаванием курса ОИВТ в средних школах.

Анализ состояния программно-методического, технического и кадрового обеспечения школьного курса ОИВТ, проведенный по результатам деловой игры «Модель выпускника средней школы по курсу ОИВТ», прошедшей среди участников 3-го Республиканского семинара заведующих и методистов кабинетов информатики областных институтов усовершенствования учителей Казахстана, а также личный опыт работы авторов в этой области позволяют высказать некоторые соображения по этому поводу.

Учитывая незначительные сдвиги в оснащении школ страны вычислительной техникой, можно все же констатировать, что положение дел остается крайне неудовлетворительным. Реальная практика обеспечения персональными ЭВМ средних школ показывает, что перспективы оснащения школ к 1990 г. более чем миллионным количеством компьютеров приобретают все более мифический оттенок. При этом мы вынуждены также считаться с тем, что существующая техника отечественного производства, приходящая в настоящее время в школу, весьма ненадежна.

В связи с этим представляется необходимым при выработке общей стратегии и тактики преподавания курса ОИВТ учитывать реально сложившееся положение. Требование достичь одинаковых результатов обучения для школ с различной материальной базой, а зачастую и с полным ее отсутствием кажется нам весьма неразумным. Вызывают также чувство неудовлетворенности учебные пособия, выпущенные разными авторами, ориентированные

на использование так называемого русского алгоритмического языка, который, по нашему мнению, больше представляет собой язык программирования, предназначенный для реализации на ЭВМ.

К этой мысли приводит имеющееся в нем излишнее количество громоздких синтаксических требований, формальное описание типов величин и др. Такая явная направленность алгоритмического языка на его перевод на язык программирования типа Паскаль или Рапира имеет смысл, на наш взгляд, только в случае обязательного выхода на ЭВМ, чего в реальных условиях мы пока повсеместно достичь не можем.

Жесткая формализация алгоритмического языка затеняет существо изучаемых понятий и решаемой задачи и тем самым превращает уроки по информатике в некую формальную и скучную игру с символами.

Мы считаем, что описанный в школьных учебниках алгоритмический язык оправданно изучать только в том случае, когда мы готовим ученика к работе на ЭВМ в качестве программиста. Однако основная масса выпускников программистами не будут, а следовательно, нет необходимости обучать всех основам программистского ремесла.

Все специальные вопросы в случае, когда исполнителем алгоритма является ЭВМ, можно было бы вынести на курсы по выбору для небольшого контингента учащихся, которые получили бы навыки программирования при более углубленном изучении курса.

Для формирования соответствующего мышления у учащихся вполне достаточен был бы описанный неформально, на уровне обычных человеческих рассуждений алгоритмический язык, например такой, как описан в книге Дж. Хьюз, Дж. Мичтом «Структурный подход к программированию» (М.: Мир, 1980. С. 80), названный авторами псевдокодом, представляющим собой текстуальное изложение алгоритма.

Таким образом, вырисовывается модель выпускника средней школы по курсу ОИВТ. Ученик средней школы должен получить представление на концептуальном уровне о составе, структуре, истории развития, поколениях ЭВМ; знать их классификацию и области применения, иметь представленные об алгоритме и его исполнителях.

В связи с тем, что подавляющее большинство будущих пользователей программно-аппаратных средств обработки информации будут иметь дело с готовыми пакетами прикладных программ, ориентированными на дружественный диалог на привычном для пользователя естественном языке с помощью иерархических меню, пиктограмм, запросов и ответов и т. д., в этом курсе нет необходимости изучать язык программирования. Работа с ЭВМ в таком виде должна напоминать привычное нам общение с бытовой техникой — телевизором, магнитофоном и др., т. е. ученик должен представлять себе возможности данной ЭВМ и имеющегося пакета прикладных программ и уметь разобраться в сопровождающей инструкции.

Все изложенное тем более важно в случае безмашинного варианта изучения курса. В этом распространенном случае необходимо на уроках по ОИВТ широко применять плакаты, фотографии, слайды, технические средства обучения, диапозитивы, кино- и видеофильмы, иллюстрирующие возможности ЭВМ, пакетов прикладных программ, существующих и перспективных периферийных устройств, способов применения программно-аппаратных средств в народном хозяйстве, роботов, автоматизированных систем управления и т. д.

Наладить тиражирование подобных наглядных пособий легче, быстрее и дешевле, чем обеспечить все школы вычислительной техникой с развитым программным обеспечением, а интерес учащихся к изучению курса резко возрастет.

Таким образом, объем курса ОИВТ при безмашинном варианте можно резко сократить без особого ущерба для существа дела за счет отказа от не имеющих в данном случае смысла формальных требований, описаний и правил алгоритмического языка.

Изучение предмета в этом случае может заканчиваться экскурсией в ближайший к школе вычислительный центр.

По мере поступления в школу персональных компьютеров с достаточной надежностью и развитым программным обеспечением можно будет приступать к ознакомлению учащихся с реальной работой с пакетами прикладных программ, представление о которых они уже получили бы на уро-

ках с применением наглядных пособий.

Кроме этого, можно было бы организовать факультативное изучение или курс по выбору для групп с углубленным изучением имеющегося языка программирования и выполнением лабораторных работ. Объем и содержание курса в данном случае будут определяться возможностями данной группы учащихся и квалификацией преподавателя.

Было бы полезным, на наш взгляд, освободившиеся часы передать в курс физики для изучения физических принципов работы ЭВМ и их применения для автоматизации научного эксперимента и математического моделирования физических и других процессов.

С появлением в школах развитого программно-аппаратного обеспечения обучение применению ЭВМ с использованием комплектов учебных и обучающих программ, программ-тренажеров, информационно-поисковых систем, программ для решения математических, физических и химических задач с элементами автоматизации научных исследований, программ для гуманитарных предметов, текстовых редакторов, компьютерных уроков по различным школьным предметам и т. д. уроком будет проводить не только на уроках информатики, но и при преподавании других предметов. При этом необходимо применять также наглядные пособия, позволяющие сообщать учащимся сведения о самых современных достижениях в области развития программно-аппаратных средств в мире, помогающие «совершать экскурсии» в самые передовые научно-исследовательские лаборатории в нашей стране и за рубежом, не выходя из стен аудитории.

Реализация новой модели выпускника средней школы по курсу «Основы информатики и вычислительной техники» предполагает написание новых учебников и учебно-методических пособий, отражающих реальное положение дела с программно-аппаратным обеспечением в средней школе и учитывающих международный опыт. Разумеется, таких учебников должно быть несколько и они должны отвечать всем возможным случаям технического обеспечения курса ОИВТ и всем потребностям национальных школ. Этим же требованиям должны отвечать наглядные пособия и программное обеспечение.

Только комплексное решение этих проблем позволит добиться решения сложной задачи подготовки поколения, готового принять вызов нового, информационного общества.

Компьютер — анестезиолог

Медики говорят: «Целительный скальпель хирурга продвигается вперед настолько стремительно, насколько позволяет ему развитие анестезиологии».

В 50-х гг. анестезиология выделилась в отдельную отрасль медицины. Служба безопасности во время операций взяла на себя многое из того, что мешало хирургам работать. Анестезиология стала наукой не только об обезболивании, но и о врачевании экстремальных состояний, нормализации остронарушенных жизненно важных функций организма. Сегодня она вступает в новую, компьютерную, фазу своего развития.

...Я стою у огромного стеклянного купола в центре демонстрационного зала Всесоюзного научного центра хирургии Академии медицинских наук СССР. Этажом ниже идет сложная операция. Прозрачное стекло позволяет наблюдать за происходящим. Вначале вижу только хирургов, другие специалисты, участвующие в операции, остаются где-то на втором плане. Взгляд не отрывается от рук в резиновых перчатках и окаймленного простынями на груди больного операционного поля.

Когда первое, самое сильное впечатление проходит, осматриваю операционную. Замечаю, что в операции принимают участие не только хирурги. Параллельно с ними работает бригада анестезиологов. И уже не знаешь, чья задача сложнее тех, кто оперирует, или тех, кто поддерживает жизнь в неподвижном теле.

Ответственный анестезиолог неотрывно следит за экраном дисплея, установленного рядом с операционным столом. Это часть новой автоматизированной системы, внедренной недавно в центре.

К телу больного прикреплены многочисленные датчики, соединенные с компьютером. На экран дисплея непрерывным потоком идет информация о состоянии всех жизненно важных систем человека, рекомендации о том, что необходимо предпринять в данную минуту.

Анестезиолог дает короткие команды. Его коллеги меняют жидкость в обступивших больного капельницах, вводят лекарства в кровь сквозь прозрачные трубки аппарата искусственного кровообращения. Как и сердце, аппарат должен четко и быстро реагировать на все физические и биохимические колебания, происходящие в организме. В ходе операции надо учитывать многие показатели. Только систему кровообращения характеризуют: артериальное и венозное давление, производительность сердца, частота его сокращений, сопротивление сосудов кровотоку, температура разных частей тела. А ведь нужно следить и за системой дыхания, работой головного мозга, других органов. Компьютер помогает решить все эти проблемы.

После операции беседую с авторами новой автоматизированной системы — руководителем отдела анестезиологии центра членом-корреспондентом АМН СССР, профессором А. Бунятыном и ведущим сотрудником отдела кандидатом медицинских наук Е. Флеровым.

— Совместно со специалистами одного из НИИ мы задумали научить компьютер не только представлять в удобной и наглядной форме необходимую для принятия решения информацию, но и давать рекомендации по применению тех или иных средств. Для работы мы использовали компьютер РДР-11/34. Это мини-ЭВМ. Емкость оперативной памяти 256К байт. Постоянная память — 2 дисковода по 14М байт. Параллельный процессор с пла-

вающей запятой. Программа написана на Фортране.

Сначала компьютер запоминал и откладывал в своей банк данных информацию о ходе 3 тыс. успешно проведенных операций и постигал логику поведения врачей. А затем его проэкзаменовали. Оказалось, что компьютер способен оказывать анестезиологу неоценимую помощь. Он мгновенно отражает на экране дисплея необходимые параметры жизненных функций пациента, записывает и хранит в своей памяти информацию о 59 характеристиках больного. Записанная на магнитные диски информация в любой момент может быть использована анестезиологом. На дисплей в операционной могут одновременно выводиться значения 12 параметров, которых вполне достаточно для того, чтобы выявить перемены в состоянии оперируемого и своевременно принять меры.

Работа с компьютером РДР-11/34 послужила полигоном для создания серийной информационно-вычислительной системы анестезиолога (ИВСА-01). Сегодня промышленность уже приступила к ее выпуску. Она включает последние образцы отечественной электроники и вычислительной техники. Базовый процессор микро-ЭВМ МС 1212 с объемом оперативной памяти в 1М байт. Цветной графический дисплей, на котором в табличном и графическом виде выявлены динамика параметров во времени. Система ИВСА-01 используется при оперативных вмешательствах на сердце. Разработано несколько вариантов ее математического обеспечения. Компьютер рекомендует, выбирает метод терапии и следит за эффективностью его применения, может взять на себя управление отдельными лечебными воздействиями.

Творческий союз «СТАТУС — XXI»
представляет

автоматизированное рабочее место учителя-словесника

«ВИРТУОЗ»

— это пакет контролирующих
программ-тренажеров (ПКПТ)
по русскому языку для 5—9 классов средней школы.

ПКПТ «ВИРТУОЗ»

предназначен для использования на итоговых
обобщающих уроках по всем разделам (темам) курса
с целью систематизации, повторения, закрепления нового материала,
проведения всестороннего и глубокого контроля знаний, умений
и навыков учащихся 5—9 классов, а также для самостоятельной
отработки учениками орфографических и пунктуационных навыков и умений.

ПКПТ «ВИРТУОЗ»

— это уникальная возможность осуществить
комплексную проверку знаний, приобретенных учащимися по любой теме
за 45—55 мин, увеличив при этом в 3—4 раза количество слов,
предложений, заданий, обеспечив 100 % занятость и резко повысив
интерес к предмету; а также возможность оперативно оценить
работу учеников по каждому выполненному заданию.

Пакет заданий, выполняемых за один урок, включает:

- 10 теоретических вопросов;
- 10 теоретико-практических вопросов;
 - словарный диктант;
 - орфографический диктант;
- морфологический (синтаксический) разбор;
 - контрольный диктант.

ПКПТ «ВИРТУОЗ» незаменимый помощник для всех,
кто преподает русский язык и изучает его.

ПКПТ «ВИРТУОЗ» — это репетитор дома.

Пакет реализован на ПЭВМ «Агат» и БК—0010Ш
и доступен любому пользователю,

в том числе и не обладающему навыками работы с ПЭВМ.

В настоящее время творческий союз «СТАТУС — XXI»
проводит работу по подготовке 2-й части ПКПТ «ВИРТУОЗ» —
«Подведем итоги за год», 3-й части — «Диктанты-марафоны» с элементами
обучения и 4-й части — «Обучающие программы 5—9 класс».

Заявки просим направлять по адресу:

125239, Москва, а/я № 61

Телефон: 383-23-35

Уважаемые коллеги!

Хочу высказать ряд соображений, которые, быть может, отличаются от мнений многих товарищей и оценок реальной ситуации со школьным предметом ОИВТ, но которые считаю важными в свете дискуссии о будущем школьной информатики.

Чтобы были понятны истоки моих суждений, несколько слов о себе. Прикладной программист с 20-летним стажем, работаю в НИИ, параллельно уже примерно десять лет обучаю школьников программированию и, как понимаю и могу, основам информатики (в рамках УПК для старшеклассников). Пробовал также силы в обучении коллег, учителей и младших школьников. Поэтому надеюсь, что меня не сочтут «человеком со стороны».

Сравниваю прожитое в мире вычислительной техники с тем, чем сегодня занимаются массы энтузиастов компьютеризации на технике уровня БК и программируемых микрокалькуляторов, и меня охватывает тоска. Неужели наши усилия и время, потраченные помимо настоящей программистской работы в большом объеме на «машинно-мышиную» возню, полностью пошли прахом и теперь уже миллионы людей, пусть на другой технике и часто дома, а не только на работе, делают, по существу, то же самое?!

Хорошо помню, как мы, исправляя ошибки, резали бритвенными лезвиями недостающие дырки в перфокартах и затыкали лишние, как читали отперфорированный текст с помощью разграфленной карты — «читалки», держа в голове или перед глазами кодовую таблицу. Аналогичное в современных условиях — это борьба с ограниченностью Фокала или Бейсика на БК, изнурительный, фактически сизифов труд по эко-

номии каждого байта из двух-трех десятков килобайт БК и ДВК, каждого шага в программе для микрокалькулятора. Если кто-то искренне считает, что такая деятельность — тоже информатика, то я полагаю, что это лишь ее грустные издержки. Мы, профессионалы и энтузиасты, должны помочь нашей электронной промышленности, поставляющей пока что негодную технику по немислимым ценам (БК, ДВК, «Агат», КУВТ-86 и т. п.), как можно быстрее избавить миллионы соотечественников от необходимости тратить интеллект на борьбу с ее ограниченностью. И особенно школьников и учителей. Надо создать общий настрой неприятия плохой техники даже в условиях ее острого дефицита.

Ощущение тоски усиливается, когда я вглядываюсь в курс ОИВТ. Особенно сопоставляя его содержание с тем, что уже сегодня доступно в нашей стране пользователям нормальных, а не удручающе примитивных персональных компьютеров. Впечатления опираются как на личный опыт, так и на материалы «ИНФО» (особенно рубрик «Методика обучения», «Педагогический опыт», информация клуба пользователей БК). Я ни в чем не хочу упрекнуть ни редакцию «ИНФО», ни энтузиастов компьютеризации, но полностью разделяю мнение*, что «поголовная компьютерная грамотность, как она понимается у нас в стране, не нужна», и теперь уже стало очевидным, что «обязательный курс ОИВТ... явился одним из уродцев...». В том, что так получилось, виноваты многие (и я, програм-

* Белошанка В., Лесневский А. Основы информационного моделирования // Информатика и образование. 1989. № 3.

мист, в том числе), неверно представлявшие еще совсем недавно пути овладения компьютерной грамотностью. Да и само ее содержание.

Чтобы обосновать такое негативное заключение, напомним, не претендуя вовсе на новизну или полноту, категории доступных сегодня на ПЭВМ программных средств, готовых к немедленному прикладному применению, т. е. для решения реально полезных задач без программирования в традиционном понимании. Два-три года назад о половине из них мы никакого понятия не имели.

1. Текстовые редакторы (процессоры) третьего поколения, как ориентированные на конкретные естественные и формальные языки, так и настраиваемые пользователем на специфические потребности текстообработки. Сюда примыкают разнообразные редакционно-издательские системы.

2. Системы управления базами данных с развитым меню-диалогом и другими составляющими «дружественности».

3. Электронные таблицы (табличные процессоры).

4. Математические ассистенты, универсальные и проблемно ориентированные программные калькуляторы.

5. Обработчики результатов измерений со встроенными цифровыми фильтрами, статистикой, графикой и т. п.

6. Графические редакторы и «рисовальщики» со встроенными «проигрывателями» для демонстрации «мультфильмов». Сюда же примыкают программы визуализации числового материала.

7. Системы автоматизированного проектирования в машиностроении, электронике, архитектуре и других сферах.

8. Экспертные системы и инструментальные средства (оболочки) для построения баз знаний и экспертных систем.

9. Музыкальные редакторы.

Если иметь в виду профессиональных программистов, то для них появились:

10. Инструментарий, облегчающий работу в операционной системе ЭВМ.

11. Языково-ориентированные текстовые редакторы, «знающие» свой язык и «умеющие» хорошо помогать при написании программы. Мощные отладчики и турбо-системы.

12. Проектировщики программ, позволяющие создание алгоритма вести с помощью ЭВМ с этапа разработки структуры или блок-схемы, а не с конечного этапа ввода готового текста на языке программирования.

Со всеми категориями средств (кроме музыкальных) автору уже удалось либо осуществить первое знакомство, либо даже всерьез

поработать. Берусь утверждать, что в совокупности они создают для пользователя принципиально иной, гораздо более совершенный уровень рабочего комфорта. Эта высокопроизводительная среда должна стать доступной всем пользователям ЭВМ, если мы действительно стремимся к настоящей информатизации, а не к пародии на нее.

Попытка даже краткого перечисления областей применения этих средств заняла бы слишком много места, равно как и педагогическое обоснование необходимости и возможности их использования во многих предметах школьной программы. Это темы для отдельного обстоятельного разговора.

Не во всем соглашаясь с упомянутой статьей, рискну по-своему сформулировать направления совершенствования ОИВТ.

1. Нет необходимости и далее стремиться всех выпускников средней школы делать (плохими) программистами. Уверен, что жизнь не заставит всех реально программировать на Бейсике или другом обычном языке программирования. Соответственно можно относительно сократить объем некоторых разделов ОИВТ.

По-видимому, нужны два курса (предмета), один из которых готовит очень грамотного пользователя ВТ и программных средств, а второй из гораздо более узкого контингента учащихся готовит программистов, будущих (после вуза) профессионалов. Около половины объема первого курса следует со временем влести в ткань других школьных предметов — математики, физики, русского языка, черчения, географии, биологии и некоторых еще. Только в этом случае можно будет рассчитывать на мощную положительную отдачу от компьютеризации образования.

2. Сформулировать на основе названных категорий программных средств немногие новые базовые понятия и принципы, отсутствующие в нынешнем курсе ОИВТ, необходимые для уверенного, правильного применения этих средств. Формулировки должны быть ближе к здравому смыслу, чем к формализмам высшей математики и математической логики, и доступны школьнику и рядовому учителю (не преподавателю ОИВТ).

3. Уделить разделу применения ЭВМ на основе освоения программных средств названных категорий гораздо больше внимания и объема в часах. Понимаю, что сделать это даже на машинах уровня «Ямаха» совсем не просто, а на технике похуже и вовсе невозможно. Совершенно необходимо улучшить технику, как это ни трудно, примерно на порядок (относительно КУВТ-86) по основным параметрам.

4. Очень невыгодно для идеи глубокой информатизации общества, что среди преподавателей ОИВТ кроме учителей физики, математики и совместителей-программистов практически нет представителей других наук и профессий. Мы неизбежно стараемся воспроизводить в массе себе подобных, тогда как обществу требуется нечто другое. Вырваться из этого замкнутого круга сложно, но

надо обязательно, и нельзя откладывать.

Надеюсь, меня правильно поймут и идеологи курса ОИВТ. Я уважаю затраченный ими труд и энергию и только предлагаю общими усилиями основательнее готовить будущее.

С уважением

А. ПОПКОВ,
канд. физ.-мат. наук

Предлагаем всем

Уважаемая редакция! Пишет вам учитель межшкольного УПК Октябрьского района г. Краснодара, в прошлом программист с 25-летним стажем. Я готовлю детей IX — X классов для работы по специальности «программист-оператор ЭВМ». В нашем МУПК есть экспериментальная группа детей III класса, пришедших в школу с 6 лет. Второй год они занимаются в классе ЭВТ, оснащенном КУВТ-86 «Бейсик». Занятия ведутся по русскому языку и арифметике. Программы составляют учащиеся старших классов, используя материал из учебников для III класса. Многие программы имеют застав-

ки в оригинальном исполнении. Детям III класса очень нравится работать с такими программами, и они с большим желанием приходят в УПК еженедельно по четвергам. Работают по схеме: 1 урок — арифметика, 2 — русский язык. Сейчас у нас готовы 40 программ по арифметике и 36 по русскому языку, продолжаем идти дальше.

Нам очень хотелось предложить наши программы всем желающим!

С уважением

Воронежская Людмила Леонидовна.

350001, г. Краснодар, ул. Пионерская, 38, Межшкольный УПК.

Как быть?

Когда было принято решение о введении в ПТУ информатики, не было ни подготовленных учителей, ни техники, и тогда мы, инженеры-программисты, пришли на помощь. На специальных курсах (у меня три удостоверения) изучали методику и начали преподавать новый предмет. Сначала я оформила кабинет для безмашинного варианта ОИВТ, договорилась с институтом об экскурсиях и каждая группа по 10 ч работала на машинах. Затем получила технику, кабинет оказался мал, оформила новый. Как оформляются кабинеты, вы, надеюсь, знаете! И ремонт, и «вся наглядность», и электрические работы и пуско-наладочные — все преподаватель, его энергия и инициатива. Два кабинета — два отработанных отпуска, чтобы к сентябрю все подготовить. В эти годы я окончила педагогический факультет ЛГПИ и теперь имею диплом о высшем педагогическом образовании. Но вот пришли новые планы — информатика объединяется с математикой, первые выпускники этой специальности закончили институт. Как быть нам? Наши места на заводах заняты, сейчас везде идет сокращение штатов. Что делать? Когда мы были нужны, нас умоляли, сорва-

ли с мест, а теперь надо оставлять налаженную работу, еще раз браться за новое дело? Не кажется ли вам, что это сталинское пренебрежение к людям — винтиков много, заменим другим? Математики, которым сейчас около 40 лет, курса не знают, с техникой работать не умеют. Выходит, наши места займут дилетанты. Вот и получается глупость за глупостью. Первая — когда предмет растянули на три года: 23 ч — на I курсе, 38 — на II, остальные — на III. Занимаясь раз в неделю, ребята забывают предыдущий материал, у них снижается интерес. Приходится придумывать факультативы, кружки, куда ходят не все. И вот новая идея — объединение курсов. Взяться за математику — боюсь, не будет качества, отдать информатику математикам — загубить предмет, по информатике экзамена ведь нет, начнут заменять математикой. Уступить место выпускнику педагогического института? Ох, далеко не все могут работать в нашей системе с отдачей.

Может, посоветуете, как быть?

Т. В. АНАНЬЕВА,
преподаватель информатики СПТУ № 1
г. Рыбинска Ярославской области

Методический кабинет института усовершенствования учителей г. Херсона получил экспериментальные программы по интегрированному курсу «Информатика и автоматизация производства на основе ЭВТ» от ВНИМЦ ПТО, которые рассчитаны на 17, 34, 57, 99 и 117 ч. В настоящее время руководствами училищ рассматривается переход на 35-часовую рабочую неделю и, естественно, возникает вопрос о том, какие предметы следует сократить.

В настоящее время на ОИВТ отводится 92 ч и на предмет «Автоматизация производства на основе ЭВТ» — 30—38 ч; итого: 122 или 130 ч (1-й и 2-й курсы). По экспериментальным программам предлагается 17 и 34 ч — ознакомительный уровень, 57 ч — пользовательский уровень (секретарь-машинистка, различные операторы, токари на станках с ЧПУ и др.); 99 ч — специальный уровень (оператор ЭВМ); 117 ч — специализированный уровень (электромеханик по ремонту ЭВМ, наладчики робототехнических комплексов и др.).

На наш взгляд, это деление неверное. Обоснуем это утверждение.

1. Почему каменщик или швея должны быть только ознакомлены, а не научены стать пользователями ЭВМ, ведь этим квалифицированным рабочим жить в XXI в.? Не секрет, что уровень подготовки наших учащихся падает. Его следует поднимать, а вместо этого мы своими же руками должны ввести дискриминацию в подготовке кадров!

2. В школах на информатику отведено 102 ч, и никто не думает сокращать часы; в олимпиадах наши учащиеся должны выступать наравне с другими. Где же логика? Наши учащиеся тоже могут поступать в вузы, мы им даем такое право, однако уровень

подготовки по предлагаемым экспериментальным программам будет более низким по сравнению с общеобразовательной школой.

3. Предмет ОИВТ переживает тяжелый период становления. Преподаватели выросли в квалифицированных специалистов, постоянно повышающих свой уровень. А с грядущими нововведениями знающие себе цену уйдут из училища. На их место придут другие, которым нужно будет время на адаптацию. У преподавателей опускаются руки — как работать в таких условиях? Ждать милости от руководства училищ, доказывать, что информатика нужна? Зачем выброшенные на ветер государственные деньги, затраченные на приобретение дисплейных классов и всего необходимого? Ведь мы отказываем нашим детям в компьютерном образовании. Куда мы придем? Будущее нам не простит, если мы сейчас промолчим. Поэтому мы настоятельно просим оказать содействие, чтобы в методических указаниях для ПТУ области установили следующее количество часов для интегрированного предмета: для групп без среднего образования (1-й год обучения) — 34 ч, для групп со средним образованием — 99—117 ч, для групп ТУ, учащихся, окончивших среднюю школу, — 34 ч.

Кроме того, мы обращаемся с просьбой о выделении ставки лаборанта для дисплейных классов.

Секция преподавателей надеется в самое ближайшее время получить положительный ответ для того, чтобы как можно скорее заняться планированием интегрированного курса на следующий учебный год.

По поручению областной методической секции руководитель секции старший преподаватель ОИВТ

Г. И. БЛИНКОВА

От редакции.

Два письма... Различны как люди, их написавшие, так и пути, приведшие их в систему профтехобразования. Но одинакова тревога за судьбу компьютеризации, за свою собственную судьбу. Ответить на вопросы читателей мы попросили завдующего лабораторией информатики и ЭВТ ВНИМЦ Гособразования СССР канд. пед. наук Эдуарда Юрьевича Красса.

Содержание нового предмета «Информатика и автоматизация производства» разработано коллективом сотрудников ВНИМЦ Гособразования СССР на основе действующих программ курсов «Основы информатики и вычислительной техники», «Автоматизация производства на основе электронной вы-

числительной техники» и программы экспериментального курса «Основы информатики и автоматизации производства» для профессионально-технических училищ и подготовки рабочих на производстве. Учтен опыт работы учебных заведений как в нашей стране, так и за рубежом.

Новый предмет является основой для дальнейшего изучения в специальных курсах вопросов, связанных с автоматизацией производства на базе электронной вычислительной техники. Вместе с тем его содержание опирается на уже имеющиеся у учащихся знания и навыки, приобретенные в ходе изучения общеобразовательных и общетехнических предметов.

Комплект учебной документации по предмету «Информатика и автоматизация производства» состоит из трех разделов: «Содержание учебного материала», «Лабораторно-практические работы», «Учебные планы и программы». Важной отличительной чертой данного комплекта учебной документации является построение учебных программ на вариативно-модульной основе, что позволяет, используя блочно-модульную структуру организации содержания учебного материала, учесть при построении вариантов учебных программ не только различную профессиональную потребность рабочего в овладении микропроцессорной техникой, но и различные сроки обучения, предусмотренные для тех или иных групп учащихся. Вариант учебной программы строится из модулей, помещенных в разделе «Содержание учебного материала», и лабораторно-практических работ, перечень которых приведен в разделе «Лабораторно-практические работы». Порядок расположения (следования) и количество модулей и лабораторно-практических работ зависят от целей обучения, определяемых уровнем изучения информатики и автоматизации производства и сроками обучения.

В разделе «Содержание учебного материала» вопросы, подлежащие изучению, зафиксированы в модулях, каждый из которых, в свою очередь, входит в один из следующих десяти блоков.

1. Информация.
2. Основы алгоритмизации.
3. Логико-математические и физические основы работы микропроцессорной техники.
4. Микрокалькулятор.
5. Компьютер. Аппаратные средства.
6. Компьютер. Программные средства.
7. Основы программирования.
8. Решение задач с помощью компьютера.
9. Средства и системы обработки и управления.
10. Профессиональная направленность курса.

Предполагается, что раздел «Содержание учебного материала» будет дополняться модулями, содержащими сведения о новой технике и технологиях, которые внедряются в отечественную и зарубежную практику производства. Естественно, что модули, со-

держащие устаревшие сведения, терминологию и стандарты, будут либо удаляться, либо заменяться. Таким образом, раздел «Содержание учебного материала» представляет собой относительно стабильную часть, изменения в которой могут происходить раз в несколько лет.

Такой же стабильной частью комплекта является перечень лабораторно-практических работ, охватывающих содержание всего учебного курса.

В разделе «Учебные планы и программы» приведены примерные образцы учебных программ на каждый из трех уровней обучения: ознакомительный, пользовательский, специализированный.

Ознакомительный уровень рассчитан на учащихся, которым в своей профессиональной деятельности на рабочем месте, как правило, не придется иметь дело с персональными компьютерами и технологическим оборудованием, автоматизированном на основе микропроцессорной техники. Цель обучения на этом уровне — дать представление об основных понятиях информатики, возможностях современных компьютеров, средствах и системах автоматизации производства на основе микропроцессорной техники.

Уровень пользователя рассчитан на учащихся, которые в своей профессиональной деятельности будут регулярно работать на персональном компьютере, или станках с ЧПУ, или другом технологическом оборудовании, автоматизированном на основе микропроцессорной техники. При этом предполагается, что они будут использовать готовое программное обеспечение, т. е. для этого уровня компьютер является средством, инструментом производственной деятельности. На этом уровне у учащихся необходимо выработать устойчивые навыки работы на компьютере или специализированной ЭВМ с использованием имеющегося программного обеспечения.

Специализированный уровень рассчитан на учащихся, для которых компьютер является объектом производственной деятельности. Им требуются глубокие знания устройства и эксплуатации персональных компьютеров, основ программирования, принципов построения и работы автоматизированных систем. В этом случае программа должна предусматривать углубленное изучение либо устройства компьютеров и автоматизированного технологического оборудования, либо основ программирования, включая умение самостоятельно разрабатывать несложные программы для персональных компьютеров или станков с ЧПУ.

Профессионально-технические училища, в

зависимости от профессии, получаемой учащимися, оснащенности рабочих мест на базовом предприятии компьютерами и оборудованием, автоматизированном на основе микропроцессорной техники, самостоятельно устанавливают для каждой учебной группы необходимый уровень обучения (ознакомительный, пользовательский или специализированный) и определяют количество учебных часов, отводимое на изучение предмета. В комплекте учебной документации приведены варианты учебных программ, которые могут удовлетворить потребности различных подготавливаемых в профтехучилищах профессий по уровню и срокам обучения. Подчеркиваю, что право выбрать вариант учебной программы из комплекта принадлежит учебному заведению.

Вместе с тем учебным заведениям предоставлено право осуществлять обучение учащихся предмету «Информатика и автоматизация производства» по другим программам, разработанным, однако, на основе модулей, приведенных в разделе «Содержание учебного материала», и с другим количеством часов, отводимым на этот учебный предмет.

Примерное рекомендуемое количество часов, отводимое на изучение информатики и автоматизации производства в соответствии с письмом Гособразованию СССР № 21—13/20—88 от 16 января 1990 г. «Об экспериментальной проверке учебных программ по предмету «Информатика и автоматизация производства» в профессионально-технических училищах», приведено в таблице:

В заключение хотелось бы отметить следующее.

1. В настоящее время идет эксперимент, в ходе которого уточняются содержательная сторона нового курса, его профессиональная направленность и оптимальное количество учебного времени, необходимого на его изучение. В период до повсеместного введения курса «Информатика и автоматизация производства» профессионально-технические училища могут работать или по действующим программам курсов «Основы информатики и вычислительной техники» и «Автоматизация производства на основе электронной вычислительной техники», или по экспериментальной программе курса «Информатика и автоматизация производства».

2. Время на изучение предмета «Информатика и автоматизация производства» определяется главным образом уровнем обучения (ознакомительный, пользовательский, специализированный), исходя из потребностей производства, оснащенности

| | Озна-комительный уровень | Уровень пользо-вателя | Специализированный уровень |
|---|--------------------------|-----------------------|----------------------------|
| 1. Подготовка рабочих в профтехучилищах из числа молодежи, окончившей базовую (основную) школу | 34—57 | 57—99 | 99—118 |
| 2. Подготовка рабочих в профтехучилищах из выпускников средней общеобразовательной школы | 17—34 | 34—57 | 42—72 |
| 3. Подготовка рабочих в профтехучилищах с вечерней (сменной) формой обучения на вечерних (сменных) отделениях профтехучилищ | 17—34 | — | — |
| 4. Подготовка рабочих в специальных профтехучилищах, в профессионально-технических училищах при исправительно-трудовых колониях, воспитательно-трудовых колониях и воспитательно-трудовых профилакториях МВД СССР | 17 | — | — |

училища персональными компьютерами, а не названием профессии, осваиваемой учащимися. Например, в зависимости от оснащенности микропроцессорной техникой рабочего места, на котором будет трудиться будущий автослесарь, он должен изучать этот курс либо на ознакомительном уровне, либо на уровне пользователя.

3. Сотрудники лаборатории информатики и ЭВТ ВНИЦ Гособразованию СССР не поддерживают объединение математики с информатикой в один учебный предмет, хотя и не оспаривают право авторов этой идеи на эксперимент.

Мы будем искренне благодарны всем, кто выскажет свои предложения по совершенствованию постановки преподавания информатики и автоматизации производства в профтехучилищах и при подготовке рабочих на производстве.

Э. КРАСС,
заведующий лабораторией информатики
и ЭВТ ВНИЦ Гособразованию СССР,
канд. пед. наук

Ассоциация создана

Государственный комитет СССР по народному образованию, Красноярский государственный университет, управление народного образования Красноярского крайисполкома провели с 9 по 14 апреля 1990 г. учредительную конференцию Ассоциации учителей информатики.

В конференции принял участие 110 человек из 8 республик СССР, 86 городов и районов страны. Среди них — 65 школьных учителей информатики, 23 преподавателя ПТУ, педучилищ, техникумов, 15 преподавателей вузов, 7 сотрудников институтов усовершенствования учителей.

Обсудив представленные документы, конференция приняла следующие решения:

1. Считать Ассоциацию учителей информатики созданной.
2. Утвердить состав правления Ассоциации.
3. Поручить Правлению приступить к формированию состава Ассоциации с оформлением членства в установленном порядке.
4. Принять в качестве рабочего документа предложенный проект устава с учетом замечаний и предложений.
5. Принять в качестве рабочего документа основные направления работы Ассоциации.
6. Утвердить план работы Ассоциации на 1990—1991 г.

Устав Ассоциации учителей информатики (выдержки)

Раздел 1. Общие положения.

Статья 1. Ассоциация учителей информатики — добровольная общественная организация, объединяющая энтузиастов информатики.

Главная цель Ассоциации учителей информатики — защита прав и интересов учителя, оказание ему всевозможной помощи, создание творческой атмосферы его деятельности.

Раздел 2. Основные задачи Ассоциации учителей информатики.

Статья 2. Основные задачи Ассоциации учителей информатики:

оказание правовой помощи, материальной и моральной поддержки членам Ассоциации, защита их творческих начинаний и авторских прав;

содействие росту научного и профессионального уровня членов Ассоциации через аспирантуру, стажировки, соискательства и командировки;

сбор и распространение информации, необходимой члену Ассоциации;

пропаганда передового педагогического опыта;

организация и поддержка экспериментальных работ по компьютеризации образования; организация учительской и научной экспертизы учебников, методических материалов, программных средств, пропаганда и распространение лучших из них, поощрение авторов.

Раздел 3. Члены Ассоциации, их права и обязанности.

Статья 3. Членство в Ассоциации.

Членство в Ассоциации может быть индивидуальным и коллективным.

Коллективными членами могут быть любые государственные и общественные организации, клубы, кооперативы, поддерживающие основные цели Ассоциации, признающие настоящий устав.

Индивидуальными членами Ассоциации могут быть отдельные лица: учителя, воспитатели детских садов, преподаватели ПТУ,

родители и т. д., непосредственно связанные с обучением детей компьютерной грамоте и применением компьютера в различных видах учебной деятельности.

Прием в Ассоциацию производится решением ее правления в следующем порядке: после получения заявки составляется удовлетворяющий обе стороны договор, подписание которого равносильно вступлению в Ассоциацию.

Статья 4. Права членов Ассоциации.

Член Ассоциации имеет право:

получать необходимую информацию, методические материалы, программные средства из фонда Ассоциации согласно договорным условиям;

получать правовую помощь, финансовую, организационную и иную поддержку Ассоциации;

участвовать в любых мероприятиях, проводимых Ассоциацией.

Статья 5. Обязанности членов Ассоциации.

Член Ассоциации обязан:

предоставлять информацию, методические и другие материалы, интересующие Ассоциацию, согласно договорным условиям;

соблюдать сроки и обязательства договоров, заключенных с Ассоциацией, о передаче материалов или о сотрудничестве.

Раздел 5. Руководство Ассоциацией.

Статья 7. Высший орган Ассоциации.

Высшим органом Ассоциации является конференция, созываемая ежегодно. Участие в работе конференции решается добровольно каждым членом Ассоциации.

Статья 9. О правлении Ассоциации.

Правление представляет Ассоциацию во взаимоотношениях с другими организациями и гражданами.

Правление избирается конференцией Ассоциации в количестве 15 человек сроком на 3 года.

Члены правления действуют на общественных началах.

Раздел 6. О комиссиях.

Статья 11. Экспертная комиссия.

Экспертная комиссия формируется правлением Ассоциации.

Основной задачей экспертной комиссии является проведение оценки программ, учебников, методических и иных материалов. Порядок проведения экспертизы определяется соответствующим «Положением об экспертизе».

Раздел 7. Правовое положение Ассоциации.

Статья 13. Ассоциация учителей информатики является юридическим лицом. В соответствии с целями ее деятельности и в предусмотренном законодательством порядке Ассоциация от собственного имени совершает

разного рода юридические акты, выступает в качестве истца и ответчика в суде и арбитраже. Ассоциация учителей информатики имеет свою печать и штамп.

Ассоциация имеет право осуществлять педагогическую, научную, просветительскую, рекламную, консультационную, экспертную, редакционно-издательскую и иную работу.

Председателем правления Ассоциации избран Евгений Васильевич Кузнецов, учитель информатики СШ № 41 г. Красноярск. Вопросами членства в Ассоциации будет заниматься Юрий Александрович Бурлев, зав. лабораторией НИИ ИВТ АПН СССР.

Основные направления работы Ассоциации

1. Создание Центра информационного, методических материалов, образцов программного обеспечения. Предполагается организовать его работу следующим образом: раз в квартал Центр выпускает каталоги всех имеющихся материалов с подробными аннотациями и рассылает их в опорные пункты Ассоциации, отдельным учителям. Учитель делает заказ по каталогу и передает его в опорный пункт или непосредственно в Центр. Центр собирает заказы, тиражирует требуемые материалы и рассылает заинтересованным лицам. Образцы методических материалов могут присылаться в Центр с мест членами Ассоциации.

2. Установление связей с организациями — разработчиками школьного программного обеспечения, получение, пропагандирование и распространение лучших программных разработок.

3. Проведение ежегодных выставок методических материалов, книг, программного обеспечения по школьной информатике совместно с Гособразованиием СССР.

4. Организация системы учительской экспертизы, учреждение памятных знаков для авторов лучших программ, брошюр, учебников, книг по информатике.

5. Создание каталога перспективных разработок и выдача целевых заказов на тот или иной программный пакет или методическое пособие на конкурсной основе.

6. Проведение ежегодной методической конференции Ассоциации, тематических или именных семинаров, курсов для начинающих учителей.

На конференции утверждается план семинаров на год и рассылается в опорные пункты ассоциации, учителям. Учитель выбирает семинар, на котором хочет побывать, и посылает заявку.

7. Организация помощи учителям, защита их прав в конфликтах с администрацией

школ, РОНО и т. д., оказание юридической помощи.

8. Составление плана экспериментальных работ по школьной информатике. Создание экспериментальных площадок двух типов. Площадки первого типа — те, где непосредственно разрабатывается содержание того или иного экспериментального курса. На площадках второго типа идет апробирование материала, полученного на площадках первого типа. Количество площадок второго типа должно на порядок превосходить количество площадок первого типа. Придание официального статуса уже ведущейся экспериментальной работе. Координация работ.

9. Совместно с Гособразованием СССР создание подсистемы персонального распределения ВТ лучшим учителям информатики, на экспериментальные площадки, под летние школы и т. д.

10. Работа со способными детьми, организация системы детских конкурсов «Юный программист» различной тематики в разных городах страны. Например, Г. Миасс проводит конкурс инструментальных средств, г. Новосибирск — конкурс обучающих программ по точным наукам, Ленинград — конкурс игровых программ и т. п. Каждый ребенок должен найти соответствующую его интересам нишу, встретиться с близкими по духу ребятами.

11. Организация и проведение летних компьютерных лагерей (региональных и местных, школьных, районных, учительских и т. п.), привлечение на них квалифицированных преподавателей, интересных людей.

12. Распространение опыта компьютерных праздников, фестивалей. Предлагается ежегодно проводить в одном из городов страны такой праздник.

13. Установление связей с международными организациями, работающими в области школьной информатики, приглашение учителей и специалистов из других стран на мероприятия Ассоциации.

14. Выпуск информационного бюллетеня, своих журналов. Члены Ассоциации должны войти в редакционную коллегию журнала «Информатика и образование» с тем, чтобы более полно осуществлять связь журнала со школой.

План работы Ассоциации учителей информатики на 1990/91 год (выдержки)

Публикация Устава и всех документов Ассоциации.

Организация отделений Ассоциации.

Сбор и распространение методических материалов, программного обеспечения.

Создание и распространение каталогов программ по пяти основным видам персональной школьной техники («Агат», «Корвет», УКНЦ, БК-0010, «Ямаха»).

Создание специализированной библиотеки по школьной информатике, каталога книг и методических материалов.

Сбор информации о ведущейся экспериментальной работе в школах страны.

Выпуск информационного бюллетеня.

Организация конференции и семинаров Ассоциации

Всесоюзная конференция по школьной информатике.

март-апрель 1991 г.

Омский педагогический институт Управление народного образования г. Омск

Семинары.

январь 1991 г.

1. Интегрированный курс «Математика—информатика—физика» Сибирский энергетический институт Управление народного образования г. Иркутск ноябрь 1990 г.

2. Программное обеспечение ПК «Агат» Методика проведения уроков г. Миасс

сентябрь 1990 г.

3. «Концептуальные основы и методика курса информатики в средней школе» Ленинградская ассоциация учителей информатики (г. Ленинград)

Проведение детских турниров и конкурсов

сентябрь 1990 г.— март 1991 г.

Очно-заочный Всесоюзный конкурс детских программ г. Чебоксары

ноябрь 1990 г.

Всесоюзный лично-командный турнир по программированию среди школ г. Красноярск

апрель 1990 г.

Третий Всесоюзный конкурс «Юный программист» (Старшие — младшим) г. Красноярск

Адрес правления:

660100 г. Красноярск, ул. Волочаевская, 44, Ассоциация учителей информатики; тел. 21-35-72.

для писем и телеграмм:

660051 г. Красноярск, а/я 11533, Ассоциация учителей информатики.

КУЗНЕЦОВ Евгений Васильевич — председатель Ассоциации, ответственный за организацию конференции, семинаров. Учитель информатики СШ № 41. 660051, г. Красноярск, а/я 11533.

СКУЛИНА Галина Федоровна — секретарь. Учитель информатики. 660036, г. Красноярск, Академгородок, СШ № 41.

АЛЕКСЕЕВА Татьяна Михайловна — куратор экспериментальных работ по информатике в школе. Учитель информатики. 456321, г. Миасс, пр. Октября, д. 25, СШ № 18.

БУРЛЕВ Юрий Александрович — ответственный за финансы, договоры, членство в Ассоциации. Зав. лабораторией НИИ ИВТ АПН СССР. 630056, г. Новосибирск, а/я 167.

ВЫМЯТНИН Владимир Михайлович — куратор по «Ямахе». Доцент кафедры физики плазмы ТГУ. 634010, г. Томск, пр. Ленина, д. 36, ТГУ, фф, кафедра физики плазмы.

ГРИЦЕНКО Алексей Николаевич — куратор по «КУВТ-86», БК-0010. Зав. кабинетом информатики. 656099, г. Барнаул, пр-т Социалистический, д. 60, каб. 15 ИУУ.

ИВАНОВА Елена Валерьевна — ответственная за организацию экспертизы. Методист. Ленинград, ул. Пархоменко, д. 13, Выборгский МУПК (РЦШИ).

ИВАНОВ Александр Михайлович — ответственный за работу с детьми. Учитель информатики. 443099, г. Куйбышев, ул. С. Разина, д. 49, СШ № 63.

КАРЕЛИНА Лариса Валентиновна — куратор по «Корвету». Учитель информатики. Чебаркуль, Челябинской обл., СШ № 1.

КОРХОВА Вера Павловна — куратор по «УК-НЦ». Учитель информатики. 195067, Ленинград, пр-т маршала Блохера, д. 44, корп. 2, СШ № 521.

КАЙМИН Виталий Адольфович — осуществляет связь с Гособразованием СССР. Зам. директора. Москва, Авиационный пер., д. 6, МГИУУ.

ЛОШКАРЕВ Сергей Алексеевич — редактор информационного бюллетеня. Зав. ИВЦ Ленинского района. 660014, г. Красноярск, пр. Красный рабочий, д. 486.

НИКУЛИНА Ирина Станиславовна — ответственная за создание фонда Ассоциации. Учитель информатики. 660100, г. Красноярск, ул. Волочаевская, д. 44. Межшкольный ВЦ.

ПАРСАДАНОВ Олег Григорьевич — ответственный за информацию, рекламу, печать. Зав. учебно-компьютерным центром. 236006, г. Калининград, Московский пр-т, д. 98. Учебно-компьютерный центр Ленинградского РОНО.

ШИПИЛЕВСКИЙ Александр Семенович — куратор по «Агатус». Учитель информатики. 428017, г. Чебоксары, ул. Павлова, д. 9, СШ № 27.

Дубна-89

Кажется, в Дубне появилась новая традиция. В это очень хотят верить строители международной компьютерной школы, которая проходила здесь с 1 по 19 августа 1989 г. Девятнадцать летних дней не были украдены у летнего отдыха детей (в этом вы убедитесь, прочитав заметку до конца), но и не были посвящены пассивному, бессодержательному времяпрепровождению.

Итак, объединились молодые ученые; их целью было провести в рамках осуществления своего проекта в Международном компьютерном клубе (МКК) детскую компьютерную школу. Спонсорами стали МКК и Дубненский машиностроительный завод.

И вот пройден путь от идеи до ее воплощения: съехались ребята — чешские, болгарские, немецкие и, конечно, наши. География советских участников была весьма обширной (Ленинкан, Ташкент, Самарканд, Якутск, Запорожье, Казань). Преподавателей в основном привлекли из ИПМ им. Келды-

ша, МАИ, МИРЭА, ОИЯИ, эти же институты помогли с техникой. Диапазон предложенных в МКШ занятий можно оценить по тематике защищенных ребятами проектов: от традиционной физики до географии и генетики. С самыми младшими детьми, которые никогда не работали на компьютере, занимались преподаватели Н. Буланова, П. Брусиловский, а также московские студенты.

Огромную работу проделал директор МКШ (он же директор СШ № 8 г. Дубны) Ю. Курлапов. Именно в компьютерном классе этой школы проходили занятия. Общежития и аудитории были предоставлены филиалом НИИЯФ МГУ. Научная программа не помешала провести интересные экскурсии, соревнования, вечера. По результатам школы в секции «Компьютер и познание» МКК подготовлен и опубликован научный отчет «Международная компьютерная школа: проблемы и перспективы».

Связанные одной цепью...

Альпинисты, намного спокойнее за свою судьбу, если они в связке. А вот компьютеры, связанные в сеть, уверенности в завтрашнем дне своим владельцам не прибавляют. И все же, помимо передачи вирусов, сети делают и немало полезной работы. Поэтому новопеченные владельцы персональных компьютеров вскоре начинают подумывать о покупке модема, который позволил бы им связаться с миром, воспользовавшись самой дешевой и доступной на сегодня телефонной линией связи.

Полнодуплексный интеллектуальный модем японской фирмы «Омрон Татейси» поможет быстро и без ошибок обменяться информацией по телефону с любым абонентом. Перед передачей модем уплотняет информацию, кодируя ее специальным образом, что позволяет передавать оцифрованные сообщения со скоростью до 19 200 бит в секунду как в синхронном, согласованном с компьютером, так и в асинхронном режиме. Такая скорость передачи текста недоступна человеку и более чем в 30 раз превышает самый быстрый темп речи. К тому же модем автоматически исправляет ошибки, внесенные помехами в телефонной сети.

Подделка? Нет, чудо техники!

В те времена, когда электроника только начинала внедряться в эстрадную индустрию, производители музыкальных инструментов, словно соревнуясь друг с другом в оригинальности, до неузнаваемости изменяли их внешний облик. Гитара из древесно-стружечной плиты, орган, уместающийся на коленях, электронные барабаны, больше напоминающие ящик для посылок...

Пришли новые времена, когда больше стали цениться красота инструмента и естественность его звучания, и у производителей электронных музыкальных инструментов появились новые проблемы.

Электронный рояль фирмы японской «Самик», несмотря на свои скромные размеры, не уступает по благодатству звучания и респектабельности клавишным

ЧТО МОЖЕТ ЭВМ

Спутник делового человека

Многое, что может понадобиться деловому человеку в дороге, сумеет сделать персональный компьютер. И даже значительно больше, чем от него может потребоваться.

Японская фирма «Шарп» предлагает карманный компьютер размером не больше обычного калькулятора, выполняющий функции переносного терминала. Его матричный жидкокристаллический дисплей позволяет отобразить несколько строк текстовой, числовой или графической информации, а полнозначная клавиатура — ввести любой текст или команду.

Помимо работы в режиме обычного калькулятора и часов, показывающих местное и любое время на планете, устройство имеет ряд дополнительных встроенных функций. Телефонный справочник, расписание движения пригородного транспорта, календарь, электронный секретарь, который вовремя напомнит о деловых встречах и поездках, — вот их неполный перечень. Кроме того, терминал можно использовать и для запоминания текстовой и числовой информации, что дает возможность поработать в дороге над статьей или путевыми заметками. А с помощью дополнительного блока памяти, вставляемого в специальное гнездо, можно увеличить ее объем до 64К байт, что позволит написать и отредактировать сразу 20 страниц текста.

С помощью этого же разъема к терминалу можно подключать блоки, превращающие его либо в энциклопедический словарь, либо в один из восьми словарей иностранных языков, либо в игровой персональный компьютер. Никаких ограничений здесь нет: очередная карта — и у вас в руках помощник в планировании семейного бюджета, еще одна — и компьютер превратился в инструктор по работе с самим собой.

Ну, а вернувшись в офис, можно с помощью стандартного интерфейса подключить автономный терминал «Макинтош» или IBM-совместимому компьютеру и передать в него накопившуюся за день информацию для дальнейшей обработки.

инструментам известных мастеров. Причем рояль поставляется фирмой в различных вариантах исполнения и отделки и позволяет воспроизвести естественные «голоса» еще 20 различных инструментов.

«Голос!»

Такую команду может подать своему компьютеру владелец японского синтезатора «Войс менаджер». И небольшая коробочка весом менее 2 кг «заговорит» с совершенно натуральной интонацией на одном из двух наречий японского языка. Синтезатор речи может быть настроен на один из пяти наиболее приятных тембров мужского или женского голоса. Кроме того, синтезатор может изменять скорость «речи» в диапазоне от 360 до 650 слов в минуту и ее громкость.

Новый синтезатор имеет словарь в 70 000 слов, который может быть расширен до 140 000 слов, и превосходит по качеству речи и другим техническим параметрам выпускавшиеся до этого подобные серийные устройства. «Войс менаджер» стыкуется с персональным компьютером по стандартному каналу и не требует специального программного обеспечения.

Цитата

«Наверное, мало кому известно, что сейчас специалисты-электронщики весьма озабочены мрачными перспективами советского компьютерного парка. Он засорен и нарастающими темпами продолжает засоряться низкокачественными ЭВМ, ввезенными из-за рубежа все теми же услужливыми посредниками».

Эти слова напечатаны в газете «Советская Россия» от 27.01.90. Хотелось бы узнать, как неназванные специалисты-электронщики оценивают качество советских ПЭВМ.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПРЕДПРИЯТИЯ «КОНТИНЕНТ»

*Совместное советско-китайское предприятие
«КОНТИНЕНТ»
предлагает вашему вниманию разработки
в области обучающего программного обеспечения.*

Наш подход к созданию обучающих программ — это научная обоснованность выбранной методики. Знания профессиональных специалистов, обогащенные навыками наших программистов, — репутация предприятия «КОНТИНЕНТ».

*Мы предлагаем пакет программ,
реализованных для IBM PC и им подобных компьютеров.*

КЛАВИАТУРА — атрибут огромного количества приборов и устройств.

КЛАВИАТУРА — почерк современности!

50 упражнений нашей обучающей программы формируют навыки работы с любой клавиатурой.

Любой шрифт, выбранная или придуманная вами конфигурация.

Слепой метод, десятипальцевая система!

Скорость печатания и отсутствие ошибок — гарантируем.

Сколькими языками владеет человек, столько жизнью ему суждено прожить.

Оригинальная программа «ИНТЕРПРЕТАТОР» — это 500 отлично выученных вами английских слов.

«ИНТЕРПРЕТАТОР» — ваш богатый словарный запас. Работа лингвиста и филолога.

«ИНТЕРПРЕТАТОР» — это ваше сэкономленное время.

Быстрый и точный устный счет — непреходящая необходимость...

Букет программ для ваших детей.

Увлекательность игры при наилучшем усвоении арифметики!

Быстрый и точный устный счет — гарантия предприятия «КОНТИНЕНТ».

Мы предлагаем пакет обучающих программ «ОКНО В ФИЗИКУ»,
реализованный для компьютеров MSX-1,2 (ЯМАХА).

Вы можете стать обладателем действительно оригинальной разработки!

Мы не подражаем — подражают нам. 17 программ-тренажеров и лабораторных работ.

Пакет «ОКНО В ФИЗИКУ» позволяет увидеть физический процесс в его развитии во времени, управлять им, получая необходимые результаты.

Обучающая система «ОКНО В ФИЗИКУ» наглядно демонстрирует эксперименты, которые сложно или невозможно поставить в учебной лаборатории.

Наша система точно передает масштабы физических величин, характерных для моделируемых экспериментов, сохраняет правильные пространственные и временные соотношения различных движений.

«ОКНО В ФИЗИКУ» — это надежный контролер знаний студентов.

Построенный в форме активного диалога, наш пакет программ развивает творческое начало, повышает интерес к физике, обеспечивает твердые знания по вопросам относительности движения, движения космического корабля, электричества, магнетизма, молекулярной физики, теории волн.

По вашему желанию пакет будет адаптирован под любую вычислительную технику.

Любые доработки по вашему желанию!

Мы стоим к вам лицом, мы слушаем вас!

Наш адрес: 125190 Москва, ул. Черняховского, 4, а/я 126

тел. 161-35-69, 124-46-75

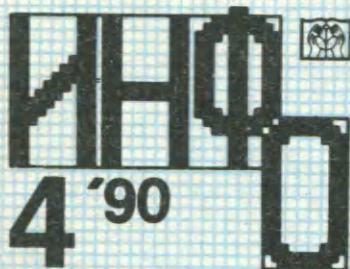
факс: 2002216, 2002217, код IRBIS-848

Журнал № 4 1990

Цена 60 коп.

70423

121



OldPC.ru
7 0 0 3
музей компьютеров

**ИНФОРМАТИКА
И ОБРАЗОВАНИЕ**

