

ИНФОРМАТИКА И ОБРАЗОВАНИЕ

5-6¹⁹⁹²

· р о с ц и о · панорама ·

ТВ
· информ ·
образование



**ХОТИТЕ БЫСТРО
ПОДГОТОВИТЬ И
КАЧЕСТВЕННО
ПРОВЕСТИ ЗАНЯТИЯ ПО
ИНФОРМАТИКЕ?**

**ВАМ поможет
Программно-методический
комплекс**

"Информатика"

**В состав комплекса входят:
ДЛЯ ПРЕПОДАВАТЕЛЕЙ:**

- Учебная программа: пояснительная записка, тематический план с указанием лабораторно-практических работ.
- Автоматизированная обучающая система «ЭСКОРТ». Предназначена для обучения преподавателей работе с классом. Имеет 2 режима работы: обучение и зачет; 2 уровня обучения: простой и сложный. АОС «ЭСКОРТ» полезна для подготовки и проведения занятий по информатике.

**Комплекс можно
приобрести целиком
или его отдельные
составные части**
Наш адрес: **107005** Имеются версии
Москва, для КУВТ-86,
Волховский УК-НЦ 01М,
пер., д. 11. УК-НЦ,
фирма «Колледж» ДВК.
тел. 267-70-58



КОЛЛЕДЖ

- Обучение преподавателей
- Переналадка старых классов
- Поставка компьютерного класса «под ключ»

**ДЛЯ РАБОТЫ
С УЧАЩИМИСЯ:**

- Автоматизированная обучающая система «КОЛЛЕДЖ». Предназначена для обучения учащихся школ и средних специальных учебных заведений по курсу «ОИ и ВТ». АОС «КОЛЛЕДЖ» избавит преподавателей от многократного повторения материала, осуществит контроль знаний учащихся. АОС «КОЛЛЕДЖ» освещает следующие вопросы: основные принципы устройства и работы ЭВМ; этапы решения задач на ЭВМ; программирование на языке BASIC.
- Сборник индивидуальных заданий для практических работ и пакет программ, используемый при изучении BASIC.
- Пакет прикладных программ.
- Уроки с программируемыми микрокалькуляторами с примерами и индивидуальными заданиями.



ИНФОРМАТИКА И ОБРАЗОВАНИЕ

Содержание

Общие вопросы

- Ершов А. Компьютеризация школы и математическое образование 3
Основные направления работ по программе «Информатизация образования» 13

Методика

- Матюшкин-Герке А. Учебно-прикладные задачи в курсе информатики 15
Хорошева И. Программа курса «Информатика» 19
Лещинер В., Матвейкина Н. Использование интегрированных пакетов 22
Ваграменко Я., Сарьян В., Прокофьев Ю., Колыхалов П., Бумагин В.,
Боловинцев Ю. «ТВ-ИНФОРМ-ОБРАЗОВАНИЕ» 28

Кабинет ВТ

- Тищенко В., Тищенко Г. Что делать, если отказал компьютер? 36
Ковтун М. Программирование в системе Express Pascal 42
Девятов А. Программирование 9-игольчатых матричных принтеров
семейства Epson 46
Григорьев С. Конфигурация для Пролога-Д 53
К-система 55
Еремин Е. Стандарт MSX для записи на МЛ 58

Клуб БК

- Эксплуатация принтера MC-6312 61
Калейдоскоп БК 61
Печать графической информации на принтерах MC-6312 и MC-6313 62
На экране поверхность 64
Джойстик? Нет проблем! 66
«Электроника БК-0010» и компьютеры, совместимые с IBM PC 67
Московский клуб БК 68
Соловьев К. Занимательные игры со строковыми функциями 69

Клуб УКНЦ

- Корнилов Г. Сетевая операционная система для УКНЦ 70
Гусев А. Операционная система ФОДОС 71
Сочнев С. Кроссворд для УКНЦ 76

Клуб «Корвет»

Ахманов С., Нечаев А., Скурихин А. Архитектура процессора KP580BM80A	80
Ахманов С., Нечаев А., Рой Н., Скурихин А. Архитектура «Корвета»	86
Дмитриев А. «Корвет» 2→1	90

Клуб «Агат»

Степанов М., Самолысов В. Дополнительные возможности графики ПЭВМ «Агат»	93
Интегрированная среда Си для ПЭВМ «Агат-9»	95

Молодежная инициатива

Олимпиады по информатике

Кирюхин В. IV Всероссийская олимпиада школьников по информатике	103
Андреева Е., Марченко А. Олимпиадная задача «Игротека» в сельской школе	105
Алексеев М. Командный турнир школьников по программированию	109

Информатика и бизнес

Юрьева Т. Бизнес: что есть что?	111
---------------------------------	-----

Нам пишут

Онегов В. Проблемные задачи на уроках информатики	115
Делекторский Н. Коллективные имитационные игры «Игротека» в сельской школе	119
К вопросу об экспертизе ППИС	121

Информация

Решение семинара-совещания	123
Конкурс ВТ	124
Фонд задач Ассоциации	125
Центр интерактивных средств обучения	12

Справочный листок

Разъясняем нормативные документы	126
----------------------------------	-----

Главный редактор
академик
В. А. МЕЛЬНИКОВ

Редакционная коллегия
И. Н. АНТИПОВ
В. Н. АФАНАСЬЕВ
И. М. БОБКО
Г. В. ГОДЖЕЛЛО
С. А. ЖДАНОВ
А. Ю. КРАВЦОВА
(зам. главного редактора)
Н. Г. МЕЛЬДИАНОВ
И. С. ОРЕШКОВ
О. К. ПАВЛОВА
А. Ю. УВАРОВ
А. И. ФУРСЕНКО
В. О. ХОРОШИЛОВ

Обложка С. Плюща, У. Громковой.
Редактор отдела Н. Копытина
Редактор отдела Н. Луцкая
Художественный редактор Л. Коновалова
Корректоры Л. Яковлева, М. Суворова

Сдано в набор 24.09.92. Подписано в печать 11.11.92. Формат 70×100 1/16.
Бумага офсетная. Печать офсетная. Усл. печ. л. 10,40. Усл. кр.-отт. 42,25.
Уч.-изд. л. 13,2. Тираж 54080 экз. Заказ 1173. Цена 4 руб.

Издательство «Школа-Пресс»

Почту направлять по адресу: 119034, Москва, Смоленский б-р, д. 4.
Издательство «Школа-Пресс». Журнал «Информатика и образование».
Телефон: 335-57-79, 275-32-23.

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

© «Школа-Пресс», «Информатика и образование», 1992

А. ЕРШОВ

Компьютеризация школы и математическое образование

Господин председатель, уважаемые коллеги!

Я беру на себя смелость обратиться к вам как к коллегам, хотя не могу считать себя ни профессиональным математиком, ни профессиональным педагогом. Тем выше я ценю приглашение программного комитета выступить на этом выдающемся научном собрании. Проработав всю жизнь в исследовательском институте и сформировав свое научное самосознание в недрах науки информатики задолго до того, как она обрела свое содержание и название, я, тем не менее, все это время предавался тому, что можно назвать математической и педагогической практикой.

Далее, имея счастливую привилегию получить математическое образование в Московском университете и общаться с выдающимися учеными за 14 лет участия в работе Отделения математики Академии наук СССР, я постоянно ощущал на себе воздействие мощных идей современной математической и образовательной мысли и ее носителей, из которых я хотел бы в первую очередь назвать Пуанкаре и Лузина, Куранта и Колмогорова, Клини и Маркова, фон Неймана и Бурбаки, Выготского и Пиаже.

Наконец, весь ход объективного развития науки информатики показывает, что математика была не только материнской наукой для информатики, но и сама информатика по мере своего становления и обособления в своих основаниях и методах неуклонно математизируется. Мало того, появляется все больше и больше свидетельств тому, что методы информатики, определенные информационные технологии проникают в глубь математики, влияют на некоторые черты стиля, техники и содержания математической работы.

Таковы три причины, которые побудили меня предложить вашему вниманию некоторые соображения о месте компьютера в математическом образовании в школе, о связи информатики и математики...

Я пытаюсь показать, что союз трех фундаментальных учебных дисциплин — языка, математики и информатики — образует нераздельную основу современного образования.

Прежде чем приступить к делу, сделаю несколько замечаний. Мы будем постоянно употреблять два весьма неуклюжих, но широко известных слова: компьютеризация и информатизация. Говоря коротко, компьютеризация — это внесение компьютеров в некоторую область деятельности человека, сопровождаемое заметной перестройкой этой деятельности под воздействием компьютера. Информатизация — это, выражаясь языком западной публицистики, построение информационного общества. В ряде публикаций эти два понятия отождествляют. В своем докладе я постараюсь в какой-то мере раскрыть эти два понятия, по крайней мере, в применении к образованию. Пока же хотел бы подчеркнуть, что отождествлять их никак нельзя и как, скажем, информатику нельзя сводить к программированию, так и информатизацию нельзя сводить к компьютеризации. Компьютер — это средство, а не содержание общественных процессов.

История изучения основ вычислительной техники и программирования, а также применение ЭВМ в учебном процессе восходит к концу 50-х годов и развивалась, примерно, по такой же схеме, как и в большинстве стран.

Реальное начало современного процесса компьютеризации школы началось в 1985 г. принятием партийно-правительственного постановления «О мерах по обеспечению компьютерной грамотности учащихся средних учебных заведений и широкого внедрения электронно-вычислительной техники в учебный процесс», которое предусматривало введение в 1985—

— Эта статья — текст последнего научного выступления А. П. Ершова на 6-м Международном конгрессе по математическому образованию в Будапеште в августе 1988 г. В журнале публикуется сокращенный вариант этого выступления, которое сегодня можно без преувеличения назвать историческим. Полный текст выступления опубликован в журнале «Программирование» АН СССР № 1. 1990.

1986 г. повсеместного изучения в IX и X классах 70-ч курса «Основы информатики и вычислительной техники» и ряд других мер, продвигающих компьютер и науку о нем в массовую школу.

Благодаря введению нового обязательного курса все пришло в движение и за последующий год было сделано больше, чем за все прошлое время. В целом это было социально-политическое решение, но за ним стоял и достаточно весомый научный багаж, подготовленный предшествующим развитием информатики и педагогической мысли. Сложившиеся определенные подходы к национальной программе компьютеризации, которые оказали воздействие на многие положения стартового постановления.

Одно лишь перечисление первых шагов, предпринятых в 1985—1986 гг., говорит о резкости старта компьютеризации.

Подготовка учебника вместе с учительским пособием и его издание пятимиллионным тиражом на 15 языках народов СССР.

Разработка программной поддержки курса информатики для доступных видов школьных ЭВМ.

Курсовая подготовка 100 тыс. учителей по информатике в 1985 и 1986 гг.

Разработка технических и санитарно-гигиенических требований к школьным компьютерам, конструирование и организация производства двух базовых моделей (архитектура типа Intel 8080 и PDP-11 LSI).

Крупная импортная закупка японских ПЭВМ архитектуры MSX.

Организация новой образовательной специальности «Информатика и вычислительная техника» в педвузах страны.

Создание в Новосибирске Научно-исследовательского института информатики и вычислительной техники Академии педагогических наук СССР.

Организация всесоюзного журнала «Информатика и образование».

Организация телевизионных уроков по информатике по всесоюзной программе объемом порядка 50 эфирных часов в год.

Организация управлений и отделов информатики и вычислительной техники на союзном, республиканском и областном уровнях и инспектората на районном уровне, выделение должности заведующего вычислительным кабинетом на уровне школы.

Хотел бы обратить внимание аудитории, что в СССР постановка школьного курса информатики проходит с явным учетом его тесной связи со школьной математикой. В основе этой связи лежит прежде всего тот социальный факт, что учителя-математики стали главным источником формирования начального корпуса преподавателей информатики: в настоящее время (ориентировочно) 55 % математики; 20 % физики; 20 % привлеченные со стороны специалисты по информатике (главным образом, инженеры); 5 % учителя других специальностей.

Это же обстоятельство повлияло на стилистику и диалектику учебника по информатике². В целом этот учебник поддерживает более или менее стандартный курс компьютерной грамотности первой ступени (правда, несколько повышенного типа, если сравнить его, например, с рекомендациями американской Ассоциации по вычислительной технике 1985 г.) Поскольку в большинстве школ регулярная работа в вычислительном кабинете невозможна из-за отсутствия компьютеров, в учебнике смещены акценты от компьютерной практики в сторону концепций, поддерживаемых задачами математического стиля и содержания (вычислительные, комбинаторные, логические и геометрические), а также некоторым посильным объемом ручной работы (возможно, с использованием карманных калькуляторов). Сами концепции (алгоритм, условие задачи, математическая модель, переменная величина, структурная организация и его систематическое построение методом последовательных уточнений и т. д.) также излагаются в алгебраизированном стиле с опорой на общематематическую символику. Эта концептуальная схема курса, естественно, поддерживается 40-ч интенсивным компьютерным практикумом там, где возможна регулярная работа в вычислительном кабинете. При этом, скажем, Бейсик трактуется не как азбука программирования, а как машинно-ориентированный производственный язык программирования, служащий для реализации ранее составленной алгоритмической спецификации, выраженной структурной алгоритмической нотацией высокого уровня.

Все эти довольно поверхностные связи базируются на гораздо более фундаментальной и постоянной связи информатики и математики. Не имея возможности подробно на этом останавливаться, я лишь только намекну на эти связи, продемонстрировав параллельный ряд концепций математики и информатики, поддерживающих важнейшее для информатики понятие исполнителя:

² Основы информатики и вычислительной техники: Пробное учебное пособие для 9—10 классов средних школ / Под редакцией А. П. Ершова, В. М. Монахова. М.: Просвещение, 1985.

Носитель.

Элемент носителя.

Операция.

Предикат.

Сигнатура.

Протокол. Последовательность операций и предикатов с их значениями + начальный элемент.

Предикат-предусловие.

Предикат-постусловие.

Допустимый протокол, реализующий на концах соответствующие предикаты пред- и постусловий.

Программа. Субрекурсивное множество, включающее множество допустимых протоколов.

Естественно, что такая связь интенсивно поддерживается и углубляется в специальной литературе. Собственно говоря, в развитии этой связи и состоит уже упомянутая математизация основ информатики. Однако с позиций образования эта связь реализуется лишь в некоторых университетских курсах и тем более еще далека от школьного образования.

Как для любой фронтальной перестройки, к тому же в значительной степени развиваемой «сверху», реальный процесс компьютеризации — это смесь успехов и компрометации, энтузиазма и фрустраций, приобщения и абсентеизма. Главные трудности — на поверхности: нехватка компьютеров, их недостаточная надежность, ненадежность распределения педагогического программного продукта, слабая подготовка учителей. Действуют и более глубокие причины: неразвитость инфраструктуры, нехватка средств, пассивность преподавателей, вызов учительскому авторитету, состоящий в том факте, что дети осваивают компьютеры, легче чем взрослые.

Но тем не менее в целом отношение учителя к компьютеризации положительное, в частности, потому, что она во многом является стержнем, двигателем и символом современного обновления школы. Кроме этого, энтузиазм учащихся, особенно при регулярном доступе к компьютеру, благотворно действует на старших. В целом компьютеризация школы набрала момент для безостановочного развития. Коэффициент полезного действия, т. е. процент учащихся, успешно осваивающих курс информатики, при всей субъективности оценен по разным регионам сравним и колеблется от оптимистической оценки в 40—50 % до реалистической в 15—20 %, сильно коррелируя с уровнем усвоения математики. Подпор выпускников средней школы с активным интересом и ожиданиями компьютеров уже начинает чувствоваться высшей школой, торопя и ее с адекватной программой компьютеризации.

Количество вычислительных кабинетов (10—15 объединенных в сеть ученических компьютеров с машиной учителя), установленных в школе к наступающему учебному году, оценивается в 2000. Разработано достаточно адекватное базовое программное обеспечение курса информатики, однако применение ЭВМ в изучении других предметов носит еще спорадический характер.

Формируется инфраструктура научной и экспериментально-педагогической работы. В стране сложилось несколько десятков центров активной научно-педагогической деятельности при педагогических институтах и некоторых университетах, в научных центрах Академии наук и институтах усовершенствования учителей.

Довольно значителен объем внеклассной работы по школьной информатике. Количество компьютерных клубов приближается к первой сотне, поставлено несколько курсов углубленного изучения информатики, очень популярны кружки по информатике для младших подростков. Складывается сеть регулярных региональных летних школ юных программистов.

Подготовлена новая версия школьного объединенного (IX и X классы вместе) учебника³, основанная на тех же идеях, что и первый учебник, но измененная в сторону большей доступности и дидактичности, учитывающая первый реальный педагогический опыт. Учебник, по-прежнему, поддерживает две методики преподавания: как с ограниченным, так и с постоянным доступом к компьютеру.

Исполнитель (робот, ЭВМ, человек в определенной роли).

Обстановка.

Состояние обстановки.

Действие, меняющее обстановку.

Вопрос к обстановке.

Система команд.

Поведение. Последовательность действий и вопросов к обстановке с ответами, отправляясь от начального состояния.

Условие задачи.

Цель задачи.

Решение задачи. Поведение, ведущее от состояния, удовлетворяющего условию, к состоянию удовлетворяющему цели.

Программа. Конечное предписание, определяющее поведение, ведущее к цели для каждого состояния, удовлетворяющего условию.

³ Ершов А. П., Кушниненко А. Г., Лебедев Г. В. и др. Основы информатики и вычислительной техники. IX—X класс. М.: Просвещение, 1988.

Одновременно состоялся открытый конкурс на альтернативный учебник информатики. Его результаты были не очень обильны, однако привели к подготовке учебника, также подерживающего концептуальное изучение информатики, более авангардистского стиля, направленного на явное изучение логико-математических основ построения алгоритмов, но, пожалуй, и более трудного для работы⁴.

Коротко можно сказать, что если объединенный учебник закладывает основы смещения курса информатики в VII—VIII классы, то конкурсный учебник «смотрит» в сторону предвузовского курса информатики.

Я не уверен, что задание XII пятилетнего плана — направить до 1991 г. в школу 450 тысяч ПЭВМ — будет полностью выполнено, однако все мы рассчитываем, что в XIII пятилетке каждая средняя школа (их у нас порядка 60 000) получит вычислительный кабинет.

За это же время курс информатики перейдет в VII—VIII классы (я здесь употребляю традиционную 10-летнюю схему обучения с 7-летнего возраста: I—III — начальная школа, IV—VIII — базовая (неполная средняя) школа, IX—X — профилирующая (вариативная) средняя школа), оставив последние два года обучения для углубленного и предпрофессионального употребления ЭВМ.

В недрах научной и экспериментально-педагогической работы созреет так называемый интегрированный курс информатики IV—VIII классов, где освоение работы с компьютером и фундаментальных понятий и умений информатики будет тесно связано со всей школьной практикой, требующей активного применения компьютеров как постоянного орудия интеллектуальной работы. Есть немало сторонников объединенного курса математики и информатики, и эта идея получит свое воплощение и проверку.

6 Регулярное применение компьютеров в начальной школе вряд ли получит полное распространение, но будет полностью подготовлено в результате опережающей совместной работы учителей и специалистов по психологии, дидактике, математике, языку и информатике.

Очень важным компонентом следующего периода будет постановка современного курса информатики и компьютерной дидактики в университетах и педагогических вузах страны. Эти курсы призваны стать опорой для формирования нового поколения учителей, которые подготовят наших детей к жизни в XXI столетии.

Одним из очень важных и положительных итогов первых лет активной работы по образовательной информатике стало формирование общей концепции компьютеризации в школе, своего рода общего взгляда на будущее человечества, на роль образования в обеспечении этого будущего, места информационных технологий и знания о них в современном образовании, попытки сформулировать долгосрочные цели, соответствующие потребностям общества и предложить возможные пути их достижения.

Информатизация общества: общие проблемы и предпосылки

1. Информатизация. Мы понимаем информатизацию как всеобщий и неизбежный период развития человеческой цивилизации, охватывающий, примерно, столетний период с 50-х годов нашего и до середины XXI века и направленный на обеспечение полного использования достоверного, исчерпывающего и своевременного знания во всех общественно-значимых видах человеческой деятельности. Информатизация, понимаемая как совокупность знаний о фактических данных и зависимостях между ними, становится при этом таким же стратегическим ресурсом общества, как и материальные и энергетические ресурсы. Техническим средством освоения такого ресурса выступают компьютеры, средства связи и другие информационные машины. При этом ресурс живого умственного труда многократно дополняется ресурсом искусственного, «машинного» интеллекта, что приводит к радикальному изменению содержания трудовых процессов.

2. Информационная картина мира. Информатизация практически реализует сложность и взаимозависимость всех жизненных процессов в совокупном организме человеческого общества, придавая этому организму своего рода центральную нервную систему. Философской и научной основой информатизации является освоение информационной картины мира, осознание единства законов функционирования информации в природе и обществе, практическое применение этих законов при создании индустрии производства и обработки информации.

Информатизация как информационный процесс состоит в обособлении и в представлении в форме, доступной для хранения, обработки и передачи электронными техническими средствами всей социально значимой информации как накопленной столетиями, так и оперативной, возникающей в процессе деятельности общества, включая познавательные про-

⁴ Каймин В. А., Щеголев А. Г., Ерохина Е. А., Федюшин Д. П. Информатика. Ч. I, II. М., МИЭМ, 1987. 228 с.

цессы отражения действительности. Не отменяя ныне существующих форм поддержки информационного фонда человечества, общество работает также в окружении программных средств, баз данных и баз знаний.

Информатизация как познавательный процесс состоит в формировании, обособлении и поддержании целостной информационной модели мира, позволяющей обществу осуществлять упреждающее динамическое регулирование своего развития на всех уровнях проявления активности — от индивидуальной деятельности до общечеловеческих институтов.

3. **Инфосфера.** Информатизация как материальный процесс состоит в строительстве инфосферы — глобальной инфраструктуры электронных средств хранения, обработки и передачи информации. Объем инфосферы для таких регионов как, например, США, Западная Европа или СССР исчисляется миллиардами микропроцессоров, встраиваемых в машины; сотнями миллионов входов в систему передачи данных и персональных ЭВМ, устанавливаемых на рабочих местах; десятком миллионов миниЭВМ, устанавливаемых в первичных ячейках народного хозяйства; сотнями тысяч крупных ЭВМ, используемых в подотраслевых и территориальных вычислительных центрах; тысячами суперЭВМ предельной производительности, замыкающих пирамиду технических средств информатизации. Их совокупная материальная ценность составляет два-три объема годового валового национального продукта региона. Удельная вычислительная мощность инфосферы в исторически обозримый период составляет 10—20 млн. операций/с на человека.

Информатизация как процесс развития производительных сил состоит в привнесении адекватной информационно-вычислительной мощности на каждое рабочее место, в резком сокращении доли живого человеческого труда в сфере высокоавтоматизированного материального и энергетического производства и радикальном повышении производительности и эффективности труда в других областях человеческой деятельности, в переходе от *технической эры* (что сделать) к *эре технологической* (как сделать).

4. **Информатика.** В нашей концепции и учебной литературе мы трактуем информатику как становящуюся фундаментальную естественную дисциплину, изучающую законы и методы накопления, обработки и передачи информации в природных, технических и социальных системах. Эта наука опирается на философское учение об информации в процессах отражения и состоит из таких конкретных областей, как теоретические основы вычислительной и коммуникационной техники, алгоритмика, программирование, искусственный интеллект, теория когнитивных процессов, включая вычислительный эксперимент, информология или учение об информационных процессах в обществе.

Наряду со становлением информатики как науки в многочисленных проявлениях складывается прикладная информатика как совокупность всех видов человеческой деятельности, связанной с использованием программных и технических средств инфосферы. Развитие прикладной информатики находит свое выражение в передаче обществу новых информационных технологий — устойчивых и общедоступных процедур систематической или автоматизированной обработки информации в интересах той или иной конкретной человеческой деятельности. Фактически, информационные технологии с самого начала сопровождают и во многом формируют человеческую цивилизацию. Здесь мы имеем в виду такие общеизвестные реалии, как изобразительное искусство, письмо, письменность, книгопечатание, почта, машинопись, телеграф, телефон, радио, телевидение, а также быстро распространяющиеся достижения последних десятилетий: ксерография, персональные вычисления, видеосистемы, оптические каналы, электронная почта, кабельное и спутниковое телевидение, настольные системы печати.

5. **Планетарное мышление человека.** Есть простая перефразировка понятия информатизации, принадлежащая Г. Р. Громову: информатизация — это совокупность взаимосвязанных политических, социально-экономических и технологических факторов, которые обеспечивают свободный доступ всем самостоятельным членам общества к любой социально значимой информации (кроме ограниченного класса законодательно временно закрытой). Действительно, в наше время право человека на информированность становится не просто нравственным идеалом, провозглашаемым просветителями и демократами, а императивом успешного развития, да и самого сохранения человеческой цивилизации, в условиях исчерпания возможностей экстенсивного развития, нарастания количества глобальных проблем, которые могут быть решены только совместно: ликвидация ядерной угрозы, сохранение окружающей среды, строительство и поддержание стабильной технической и социально-экономической мировой инфраструктуры. Общая тенденция научно-технического прогресса такова, что труд человека приобретает все более индивидуальный и творческий характер, а его влияние на окружающий мир выходит далеко за пределы его рабочего места. Нет возможности останавливаться на этом важнейшем положении более подробно, замечу только, что на стыке тысячелетий совокупная энергия мыслительной дея-

тельности человека приобретает поистине планетарный характер, о чем пророчески говорил великий русский естествоиспытатель академик В. И. Вернадский.

Образовательная информатика: предпосылки и ожидания

1. Педагогический взгляд на компьютер. Для многих педагогов характерна охранительная реакция на компьютер, подчеркивающая «орудийность» этого технического изделия. Эта охранительная реакция иногда выдается за правило хорошего тона, демонстрирующего просвещенный подход к компьютеризации школы.

Не посягая на достоинство педагогической науки, хотел бы подчеркнуть, что сорокалетний опыт применения электронной вычислительной техники позволяет дать «формулу компьютера», перечислив его наиболее продуктивные свойства: доступность, производительность, универсальность, программируемость, адаптируемость, развиваемость, сочетаемость, глобальность, идеальность.

Эти свойства, хотя они и общеизвестны, заслуживают своего анализа. Однако сейчас, пожалуй, более уместно показать, как эти свойства преломляются в педагогических ожиданиях, адресованных компьютеру. Имеется достаточный опыт, позволяющий эти ожидания сформулировать следующим образом.

1. Компьютер является наиболее адекватным техническим средством обучения, поддерживающим деятельностный подход к учебному процессу во всех его звеньях: потребность — мотивы — цель — условия — средства — действия — операции.

8 2. Будучи в состоянии принять на себя роль активного партнера с динамическим сочетанием вызова и помощи, компьютер тем самым стимулирует активность учащегося.

3. Программируемость компьютера в сочетании с динамической адаптируемостью содействует индивидуализации учебного процесса, сохраняя его целостность.

4. Контролируемость учебного процесса в сочетании с гибкостью и разнообразием пользовательского интерфейса делает компьютер идеальным средством тренировочных стадий учебного процесса.

5. Внутренняя формализованность работы компьютера, строгость в соблюдении «правил игры» в сочетании с принципиальной познаваемостью этих правил способствуют большей осознанности учебного процесса, повышают его интеллектуальный и логический уровень.

6. Способность компьютера к построению визуальных и других сложных образов существенно повышает пропускную способность информационных каналов учебного процесса.

7. Компьютер вносит в учебный процесс принципиально новые познавательные средства, в частности вычислительный эксперимент, решение задач с помощью экспертных систем, конструирование алгоритмов и пополнение баз знаний.

8. Являясь ведущим и массовым инструментом научно-технической революции, компьютер самим фактом органического включения в учебный процесс сближает сферу образования с реальным миром.

9. Наконец свойства универсальности и программируемости, способность к многоцелевому применению компьютера позволяют во многих случаях сократить стоимость технических средств обучения за счет исключения затрат на натурные эксперименты и лабораторные работы и более дешевой программной настройки с одного применения на другое.

Огромное разнообразие ролей компьютера в учебном процессе в своей основе является сочетанием трех главных функций компьютера в учебном процессе:

компьютер как орудие,

компьютер как партнер,

компьютер как источник формирования обстановки.

2. Меняющаяся роль учителя. Педагогика сотрудничества, деятельностный подход к учебному процессу, активизация учащихся, индивидуализация обучения — все эти нарастающие парадигмы современного образования радикально меняют роль учителя в классе. Авторитарная схема синхронного управления классом из нескольких десятков человек при всей ее экономности и кажущейся эффективности все больше себя изживает.

Это за пределами моей компетенции — подробно анализировать возникающие здесь педагогические альтернативы. Хочу только с уверенностью сказать, что компьютер может весьма значительно помочь учителю в определении его нового места в классе.

Хотелось бы предостеречь от ошибочного взгляда на компьютер как на препятствие, разрушающее контакт учителя и ученика. Некоторые учителя представляют компьютерный урок как дилетантское состояние, когда все смотрят на дисплей и клавиатуру, не видя друг друга, а учитель кроме контроля за классом должен дополнительно взвалить на себя хлопоты по манипулированию своим компьютером. Такое представление основано на некачественном и прямолинейном подходе к компьютерному уроку. На самом деле компью-

тер предоставляет учителю большой резерв поддержки, делающий его отношения с учениками даже более человечными, чем ранее.

Во-первых, компьютер замыкает на себя большую часть контрольных функций и реакций на ошибки ученика. Ошибки ученика беспощадно фиксируются компьютером, но становятся в значительной степени его частным делом, а не предметом получения отрицательных стимулов от учителя. Учитель становится более свободен и более позитивен в своем отношении к детям.

Во-вторых, компьютер, вступая с учеником в партнерские отношения, освобождает учителя от необходимости поддерживать темп и тонус деятельности ученика. В классе больше событий происходит «сами собой», в результате чего учитель получает больше возможностей видеть обстановку в классе в целом или уделять разовое внимание отдельному ученику.

Естественно, однако, что все это реализуется при хорошем техническом, программном и методическом обеспечении урока, а сам учитель должен достаточно непринужденно и свободно владеть общими навыками работы с компьютером.

3. Номенклатура общеобразовательной информатики. Ограничительный подход к компьютеризации проявляет себя по многим показателям. Один из них — видение компьютеризации просто как физической установки компьютеров в школах, внедрения курса компьютерной грамотности в учебный план и первоначального преодоления барьера доступности к компьютеру со стороны учителя. Считается, что после этого все образуется методом саморазвития.

На самом деле образовательная информатика — это обширная номенклатура многих видов деятельности, пронизывающих всю систему образования в целом. Прежде всего образовательную информатику составляют следующие виды «оконечной» деятельности по применению компьютеров в школе:

предметное применение, т. е. непосредственное изучение информатики как науки и компьютера как устройства;

учебное применение, помогающее изучению остальных преподаваемых предметов и дисциплин;

орудийное применение вычислительных средств и информационных технологий для поддержки универсальных видов учебно-познавательной деятельности: письмо, счет, коммуникация, графика, накопление и организация информации;

трудоовое применение, т. е. органическое использование вычислительных средств и информационных технологий при выработке трудовых навыков или освоении профессии;

досуговое, охватывающее все виды применений, связанных с реализацией личных интересов за рамками регулярного учебного процесса;

дефектологические, т. е. все виды применений, ориентированные на контингент учащихся, обладающих недостатками развития или формами инвалидности;

преподавательское применение вычислительных средств и информационных технологий учителями и преподавателями; и наконец,

организационное применение, охватывающее всю организационную и управляющую структуру системы образования.

Каждое из этих окончательных применений поддерживается соответствующей номенклатурой обеспечивающих видов деятельности:

научное обеспечение, формирующее сам предмет информатики как образовательной дисциплины и подвергающее анализу возможности и пределы применения вычислительных средств в образовании на основе синтеза соответствующих разделов педагогики, психологии, когнитологии, физиологии, медицины и социологии;

учебно-методическое обеспечение, включающее находки и рекомендации научного обеспечения в виде учебно-методических комплектов: учебников, методических руководств, педагогических программных продуктов, вспомогательной литературы и наглядных пособий;

техническое обеспечение, включающее вычислительную технику, средства связи, аудио- и видеотехнику и т. д.;

организационное обеспечение, поддерживающее всю инфраструктуру образовательной информатики и ее связь с производителями вычислительных средств, издателями, попечительскими советами и т. д.;

кадровое обеспечение, включающее профессиональную подготовку учителей по информатике, базовую подготовку по информатике всего корпуса преподавателей и организаторов образования и педагогическую ориентацию высших специалистов, вовлекаемых в сферу образования в связи с компьютеризацией.

Таким образом, реальное содержание образовательной информатики — это полноразмерная матрица порядка 8×5 , каждая клеточка которой насыщена интенсивной и постоянной работой.

4. От компьютерной грамотности к информационной культуре. Говоря о компьютеризации образования, нельзя не думать об общем направлении этого процесса, о его развитии во времени, как при переходе учащегося из класса в класс, так и при движении общества в целом вдоль оси исторического развития.

Начальный этап компьютеризации практически повсеместно во всех странах связывается с достижениями того или иного уровня компьютерной грамотности. Этот термин находится в постоянном употреблении уже почти десять лет, и я не буду анализировать его варианты, но дам его расширенное определение, которое позволило комиссии по программе перспективного учебника по информатике дать формулу развития компьютеризации школы как перехода от компьютерной грамотности учащихся к информационной культуре общества.

Курс «Основы информатики и вычислительной техники» должен формировать у учащихся:

навыки грамотной постановки задач, возникающих в практической деятельности, для их решения с помощью ЭВМ;

навыки формализованного описания поставленных задач, элементарные знания о методах математического моделирования и умение строить простые математические модели поставленных задач;

знания основных алгоритмических структур и умения применять их для построения алгоритмов решения задач по их математическим моделям;

понимание устройства и функционирования ЭВМ и элементарные навыки составления программ для ЭВМ по построенному алгоритму на одном из языков программирования высокого уровня;

навыки использования основных типов информационных систем и прикладных программ общего назначения для решения с их помощью практических задач и понимания основных принципов, лежащих в основе функционирования этих систем;

умение грамотно интерпретировать результаты решения практических задач с помощью ЭВМ и применять эти результаты в практической деятельности.

Эти требования, взятые в их минимальном объеме, составляют задачу достижения первого уровня компьютерной грамотности, а в максимальном объеме — перспективную задачу — воспитание «информационной культуры».

Возможно, на внешний взгляд это определение кажется слишком программистским, но оно достаточно емкое и оставляет простор для многих интерпретаций, чтобы по мере накопления опыта усиливать гуманитарную и социальную компоненты информационной культуры.

Воздействие компьютеризации на математическое образование

Прежде всего замечу, что математики тоже люди и им компьютер может помогать непосредственно, как и всем остальным. Я вижу, как все большее и большее число математиков, в том числе и весьма почтенных, осваивают персональные компьютеры и рабочие станции с энтузиазмом подростков. Математики хорошо думают и в этом пока прекрасно обходятся без искусственного интеллекта, баз знаний и экспертных систем. Однако компьютер дает им число и образ — и это многим очень помогает.

Раскроем теперь обозначенную тему более систематично.

1. Резкое расширение математической практики. Повсеместное применение компьютеров, строительство информационной модели мира раздвинули объем и разнообразие математической практики в грандиозных масштабах. Многие инструменты и методы математической работы становятся буквально общим достоянием. Построение знаковых систем, схематизация конкретных объектов путем выделения их свойств, атрибутов и отношений, построение моделей, дедукция, редукция и рекурсивное мышление, выделение и поддержание уровней абстракции, прогнозирование поведения, анализ законов, установлений и правил, наконец, конструирование огромного количества алгоритмов и их оценка — все это становится оружием современного интеллекта, каркасом информационной культуры. Таким образом, компьютеризация является и средством, и выражением экспансии математического знания, и этот общемировой процесс не может оставаться незамеченным самой математикой.

2. Изменение номенклатуры математических знаний. Компьютер имитирует или, лучше сказать, воспроизводит человеческое поведение. Через программирование и построение информационных моделей в содержательную часть математики входят абстракции человеческой деятельности, свойства искусственных и живых (биологических и социотехнических) систем. Все это резко усиливает роль и место дискретной математики. Этому же содействует сдвиг в физике в сторону квантовых свойств материи. Появляются разделы дискретного анализа, так сказать, параллельные классическому. Выходит на пер-

вый план изучение связи между дискретным и непрерывным, как, например, в синергетике и теории катастроф. Появляются новые приемы математической работы, например доказательные вычисления. Появляются и требуют философского осмысления такие неортодоксальные математические доказательства, как шумевшее на весь мир решение проблемы четырех красок.

3. Системная роль математической теории. Понятие теории родилось в недрах математики. С другой стороны, в информатике имеется важнейшее понятие обстановки. Обстановка — это воплощенная в компьютере замкнутая модель мира, в которой предстоит действовать программируемому исполнителю. Поскольку все исходы поведения исполнителя должны быть предсказаны, необходимо на практике обладать полным знанием обстановки и заодно сознавать пределы этого знания в реальном мире. Все это знание должно предшествовать конкретному конструированию. Выработка этого знания составляет сущность системного анализа, а построение теории обстановки становится его итогом. Системный анализ — это новый массовый вид человеческой практики, в которой роль математического метода чрезвычайно велика.

4. Вычислительный эксперимент с математической моделью. Его роль в инженерной практике общеизвестна, и я позволю себе не развивать этого тезиса. Его практичность как нового метода познавательной деятельности в учебном процессе также подтверждается педагогической практикой. Хотелось бы подчеркнуть, что в последние годы вычислительный эксперимент все в большей степени становится источником чисто математических открытий. Здесь я хотел бы сослаться на интересный обзор такого рода находок, сделанный д-ром Хазевинкелем из амстердамского Центра математических и вычислительных наук.

5. Визуализация абстракций. Визуальное восприятие человека является поистине магическим кристаллом, позволяющим делать открытия. Поиски того как сделать мысль наглядной, всегда были мучительным делом ученых и воспитателей. Интеллектуальная графика имеет тысячелетнюю историю — от наскальных изображений каменного века до гравюр Эшера. И здесь компьютер со своей способностью к синтезу изображений помогает человеческой пронизательности. Многие из нас читают и просматривают журнал «Сайнтифик Американ» или его русский вариант «В мире науки». За последние годы добрая треть публикуемых в нем уникальных иллюстраций — это образы, порожденные абстрактным знанием, оживленным союзом ученого, программиста и компьютера.

В педагогических сочинениях академика А. Н. Колмогорова неоднократно подчеркивается различие между формальной символической системой и содержательным математическим знанием — различие, игнорируемое адептами взгляда на математику как на синтетическую теорию, но хорошо ощущаемое каждым живым математиком. Думается, что аналогичная пропорция должна соблюдаться и на образовательном уровне между абстрактным математическим объектом и его визуальной моделью; пропорция, хорошо известная для геометрических объектов, но на самом деле воображая (и реализуемая с помощью компьютера) и для несравненно большего запаса математических идей. Насколько же важен яркий видимый образ для активизации молодого ума — это хорошо известно каждому учителю, воспитателю и психологу.

6. Динамизация математических объектов. Математика — это наука об инвариантах. Познать природу инварианта можно, однако, только осознав диалектику постоянства и изменчивости параметров этого инварианта. Как сказал Карл Маркс: «Любой закон проявляется при попытке его нарушения». Увидеть в логической константе все проявления реальной жизни, описанной законом, — это значит понять закон и научиться его применять. Компьютер со своими средствами визуализации и вычислений позволяет наблюдателю извлечь из статической упаковки математического отношения всевозможные траектории развития динамического процесса как во времени, так и в пространстве, обогащая тем самым его опыт, интуицию и способность к предсказанию. Все это приближает учебный процесс к исследованию и эксперименту.

7. Становление структуры из хаоса. Среди возможностей, предоставляемых математическим экспериментом и способностью компьютера к визуализации, заслуживают особого упоминания эксперименты по наблюдению становления регулярных структур из исходного беспорядка. В их простейшем проявлении — это разнообразные конструкции, возникающие при итеративном применении некоторых нелинейных операторов к случайным исходным данным или попутным параметрам. Хочу лишь привлечь внимание к тому, что образовательный потенциал этой области математических явлений еще используется в очень малой степени. Здесь формируется совершенно новый и исключительно мощный канал для распространения математического познания на огромный класс природных явлений: движение материков, формирование береговой линии, горные ландшафты, рисунок полярных

сияний, формообразование у растений, расцветка животных, развитие конфликтов и возникновение кризисов.

Стати говоря, материал, поставляемый синергетикой и математикой нелинейного, позволяет сделать важный образовательный вывод о принципиальной важности вычислительного эксперимента как познавательного инструмента: если источником всего нового в природе является нелинейность, то умозрительное предсказание экстраполяционного типа линейно по своей природе и поэтому ограничено в своей познавательной силе, как, например, любой вывод в существующей аксиоматике. Поэтому для добычи поистине нового знания необходим нелинейный синергетический процесс либо в мозгу у человека, либо в памяти компьютера.

8. Воспитание базовых способностей и умений. В популярных изданиях по математике мы постоянно находим такие оговорки: «Эта книга не требует от читателя никаких специальных знаний, но предполагает некоторую привычку к абстрактному рассуждению». Жаль, что о таких вещах мы обычно читаем лишь в предисловиях, а говорим лишь в коридорах. А между тем возможно, что это предметное «коридорное» образование и есть тот самый истинный стержень математической культуры, составляющей главную задачу школьного математического образования.

Я хотел бы подтвердить мысль, высказанную уже многими, хотя бы С. Пейпертом о том, что компьютер с предоставляемой им возможностью прямого манипулирования с визуальными образами математических объектов в искусственных мирах может сделать эту задачу предметного (или математического?) образования предметным компонентом учебного процесса, особенно в начальном обучении и в младшеподростковом периоде. Это и логические задачи, и соревнования по устному счету, составление и соблюдение «правил игры», конструирование искусственных миров, прямое манипулирование с математическими объектами, управление исполнителями, планирование их деятельности и многое другое.

9. Пробуждение первичного интереса. Продолжая только что высказанную мысль, добавим, что динамический, наглядный, послушливый и стимулирующий стиль поведения компьютера в сочетании с его универсальностью делает его идеальным инструментом для пробуждения начального интереса к математике, к ее красоте, неожиданности, предсказательной силе и волшебной связи со всем окружающим.

Братский союз математики, информатики и лингвистики

Я рискну дать еще одно, ограниченное, но очень важное, как мне кажется, определение информатики: информатика — это наука о правилах целеустремленной деятельности. Это определение становится просто справедливым, если мы уравнием информатику с наукой о компьютере (computer science — как и говорят в англоязычных странах) и если мы признаем компьютер в соответствии с тезисом Черча и тьюринговым понятием универсальной машины (с оракулом) всеобъемлющей моделью целеустремленной деятельности.

Если мы хоть отчасти согласимся с этим, то сразу обнаружим, что новорожденная информатика по праву входит в братский (если по-русски, то сестринский) союз с математикой и лингвистикой, закладывая в школьное образование опорный треугольник развития главных проявлений человеческого интеллекта: способность к обучению, способность к рассуждению и способность к действию.

Дисциплина действия так же нужна человеку, как дисциплина ума и дисциплина речи. Упражняясь в управлении компьютером, человек вырабатывает способность управлять собой. Понимая, как компьютер решает задачу, он сохраняет это понимание в себе. Наблюдая катастрофы в искусственных мирах, он многократно и безопаснее для себя вырабатывает опыт сопоставления решений и их последствий. Иными словами, я верю, что информатика позволит преодолеть созерцательность, рефлексивность и некоторый инфантилизм современного интеллектуального образования школьников. Возможно, такой вывод покажется кому-нибудь слишком максималистским, да и не хотелось бы представлять компьютер как панацею, но если мне удалось разделить с вами хотя бы долю энтузиазма, который движет школьную информатику, я могу считать свою задачу выполненной.

Основные направления работ по программе «Информатизация образования»

1. Информатизация обучения

1.1. Психолого-педагогические проблемы информатизации обучения.

1.1.1. Психолого-педагогические проблемы информатизации дошкольного образования.

1.1.2. Психолого-педагогические проблемы информатизации общего среднего образования.

1.1.3. Психолого-педагогические проблемы информатизации профессионального образования.

1.2. Информатика как предмет обучения.

1.2.1. Учебно-методические комплексы по начальной компьютерной грамотности.

1.2.2. Учебно-методические комплексы профильных и факультативных курсов информатики.

1.2.3. Учебно-методические комплексы по базовым курсам основ информатики.

1.2.4. Учебно-методические комплексы компьютерноориентированных курсов для профессиональной подготовки учащихся по специальностям, связанным с применением ИТ.

1.2.5. Учебно-методические комплексы компьютерноориентированных курсов для самостоятельного освоения учащимися новых информационных технологий (ИТ).

1.2.6. Информационные базы учебно-методических и дидактических материалов, компьютерных программ, рекомендуемых для использования при обучении информатике.

1.3. Информатизация как средство обучения.

1.3.1. Новые информационные технологии для повышения качества обучения общеобразовательным предметам.

1.3.3. Компьютерноориентированные курсы общеобразовательных предметов для самостоятельного изучения учащимися.

1.3.4. Критерии оценки компьютерных программ, другого учебно-методического обеспечения, поступающего на образовательный рынок.

1.3.5. Информационные базы учебно-методических материалов, компьютерных программ, рекомендуемых для использования при обучении общеобразовательным предметам.

1.3.6. Развивающие и игровые программные средства.

2. Информатизация управления образованием.

2.1. Информатизация работы управляющих подразделений системы образования.

2.1.1. Аппаратное и программное обеспечение применения ИТ в работе управленческих кадров системы образования.

2.1.2. Информационное обеспечение управляющих подразделений системы образования.

2.2. Создание общего информационного пространства разработчиков и потребителей ИТ в сфере образования.

2.2.1. Развитие международной электронной почты для обслуживания сферы образования.

2.2.2. Системы компьютерной связи.

2.2.3. Информационные базы данных по кадровому и материально-техническому обеспечению информатизации образования.

2.3. Информатизация ведения научных работ.

2.3.1. Программное и информационное обеспечение конкурсного и грантового финансирования научных исследований.

2.3.2. Информационные базы научных исследований в образовании.

2.3.3. Анализ и прогнозирование развития ИТ в сфере образования.

3. Информатизация в области работы коррекционных образовательных учреждений.

3.1. Учебно-методическое обеспечение коррекционных образовательных учреждений.

3.1.1. Психолого-педагогические проблемы информатизации работы коррекционных образовательных учреждений.

3.1.2. Требования к ИТ, нацеленные на применение в коррекционных образовательных учреждениях.

3.1.3. ИТ обучения для коррекционных образовательных учреждений.

3.2. Информационное обеспечение коррекционных образовательных учреждений.

3.2.1. Информационные сети коррекционных образовательных учреждений.

3.2.2. Банк данных ИТ и аппаратных средств, рекомендуемых к использованию в коррекционных образовательных учреждениях.

4. Подготовка кадров

4.1. Подготовка педагогических кадров.

4.1.1. Психолого-педагогические проблемы информатизации в системе подготовки педагогических кадров.

4.1.2. Учебно-методические комплексы для информационной подготовки студентов педагогических вузов.

4.1.3. Учебно-методические комплексы для подготовки преподавателей информатики.

4.1.4. Учебно-методические комплексы для подготовки учителей-предметников к использованию ИТ в обучении.

4.2. Переподготовка педагогических кадров.

4.2.1. Учебно-методические комплексы для переподготовки специалистов к использованию ИТ в своей работе.

4.2.2. Учебно-методические комплексы самостоятельной подготовки педагогов к использованию ИТ в своей работе.

4.3. Информационное обеспечение подготовки педагогических кадров.

4.3.1. Сети информационного обеспечения работы учреждений подготовки педагогических кадров.

4.3.2. Базы данных учреждений подготовки педагогических кадров.

4.4. *Переподготовка взрослого населения.*

4.4.1. Учебно-методические комплексы подготовки управленческих кадров сферы образования к применению НИТ.

4.4.2. Учебно-методические комплексы переподготовки взрослого населения по специальностям, с использованием НИТ.

4.4.3. Применение НИТ для переподготовки временно нетрудоустроенного взрослого населения.

Структура заявки научно-исследовательского проекта на участие в программе «Информатизация образования»

1. Информационная характеристика научно-исследовательского проекта.

1.1. Наименование научно-исследовательского проекта.

1.2. Автор заявки проекта (указать фамилию, имя, отчество, почтовый адрес, телефон, место работы).

1.3. Научный руководитель работ по проекту (указать фамилию, имя, отчество, научную степень, звание, место работы).

1.4. Исполнители (определить форму организации исследовательского коллектива: институт, НИС или кафедра института, отдел, лаборатория, ВНИК, отдельный автор и др.). Указать почтовый адрес и телефон.

1.5. Шифр проекта (указать наименование программной области, подпрограммы, направления исследования, в рамках которого разрабатывается данный проект).

1.6. Организация, представившая заявку проекта (указать ее полное наименование, почтовый адрес, телефон).

1.7. Сроки работ по проекту.

1.8. Этапы работ и ожидаемые результаты по каждому этапу в виде, допускающем их экспертную оценку.

1.9. Перечень научной продукции по результатам работ по проекту на ближайший финансовый год.

2. Научное обоснование научно-исследовательского проекта.

2.1. Проблема исследования (провести анализ ситуации в данной области, раскрыть, в чем состоит актуальность исследования, какие противоречия в сфере образования обуславливают его необходимость, сформулировать научную проблему исследования).

2.2. Объект и предмет исследования, цель, задачи и гипотеза исследования (раскрыть общий замысел проекта, его ведущую идею, описать, что предполагается исследовать в работе, указать основные методы исследования, предполагаемые результаты, условия и способы их достижения).

2.3. Практическая значимость ожидаемых результатов (определить характер и границы изменений существующей практики образования, степень готовности предполагаемых результатов для использования в практике).

2.4. Новизна исследования (раскрыть, в чем принципиальное отличие ожидаемых результатов от уже имеющихся в практике образования аналогов).

3. Технико-экономическое обоснование проекта.

3.1. Задел группы исполнителей в данной области (охарактеризовать конкретные полученные ранее результаты, указать основные публикации, время работы в данной области).

3.2. Имеющееся ресурсное обеспечение группы исполнителей:

материально-техническое обеспечение (указать наличие помещений, множительной техники, ЭВМ, издательско-полиграфических возможностей, программного обеспечения, приборов и установок);

информационное обеспечение (указать наличие банка данных в исследуемой области или возможность доступа к нему);

кадровое обеспечение (указать состав группы, представить ее научный потенциал);

финансовое обеспечение (описать, из каких источников или фондов, в каких объемах финансировалось или финансируется данное исследование).

3.3. Требуемое ресурсное обеспечение проекта на 1992 год:

материально-техническое обеспечение,

кадровое обеспечение,

финансовое обеспечение, в том числе:

фонд заработной платы, фонд на оборудование, иные расходы.

Заявки на участие в конкурсе, оформленные в соответствии с вышеизложенными требованиями, просьба присылать в четырех экземплярах по адресу:

101856, Москва, Чистопрудный бульвар, д. 6. Министерство образования Российской Федерации, Главное управление развития и координации научных исследований.

На конкурс проектов по Государственной научно-технической программе «Информатизация образования».

Срок — до 15 января 1993 года.

Телефон: 924-87-47.

А. МАТЮШКИН—ГЕРКЕ

Учебно-прикладные задачи в курсе информатики

Приводимые ниже задачи скомпонованы в тематические группы. Один из апробированных на практике подходов к их использованию состоит в выдаче рассчитанных на 2—3 месячный срок групповых (бригадных) заданий на группу учащихся из 3—5 человек с последующей защитой этих заданий.

Особо подчеркнем, что для решения этих задач не требуются какие-то особо мощные ПЭВМ (как, впрочем, и для большинства других разделов курса!). Многие из них вполне могут быть решены (и фактически решались) даже с помощью обычных программируемых калькуляторов. Чересчур ретивым пропагандистам построения учебного процесса исключительно на базе новейших моделей профессиональных компьютеров и особенно — их слишком уж доверчивым слушателям следует это учесть!

Перед тем, как завершить эту статью списком задач, сделаем одно замечание в отношении межпредметных связей курса информатики. Конечно, задачи, подобные приводимым здесь, должны были бы решаться главным образом в курсах физики, математики, биологии и т. д. Но там они должны представлять собой органически цельную часть курса. Ведь подлинное, а не показное использование компьютеров в преподавании любых дисциплин предполагает глобальную ревизию как их содержания, так и методики преподавания. Для этого, в свою очередь, нужно и хотя бы минимальное техническое оснащение учебного процесса (чего пока мы, как правило, не имеем), и должным образом подготовленные учителя (их сейчас очень мало), и новая учебная литература, и разработанное с ориентацией на новое содержание курсов (а не на сегодняшнее, как это сейчас повсеместно делается) их программно-информационное обеспечение.

То, что предлагается здесь, — это всего лишь «окошечки» в иные школьные дисциплины. Но — отнюдь не в те, которые существуют сегодня! Это «окошко» в будущий курс физики, в будущий курс математики, в будущий курс биологии и т. д. Приводимые здесь задачи предназначаются, конечно, не для то-

го, чтобы сколько-нибудь существенно дополнить какой-либо из сегодняшних курсов. Помимо той методологической нагрузки, о которой только что шла речь, их изучение закладывает основы единого подхода к разработке и созданию этих будущих курсов, в определенной мере может служить ориентиром для педагогов-экспериментаторов.

Подобно тому, как использующаяся в подавляющем большинстве вузовских дисциплин математика представлена, тем не менее, самостоятельным предметом в институтских учебных планах, так и школьная информатика останется безусловно необходимой и тогда, когда весь учебный процесс будет (по-настоящему!) переведен на электронные рельсы.

Останутся нужными (вопреки безапелляционным утверждениям иных горе-теоретиков) и сам курс, и рассмотренный здесь раздел. Ведь исключив его, мы создадим предпосылки к появлению такого вавилонского столпотворения различных подходов, терминологии, методов даже в рамках одной и той же учебной дисциплины, не говоря уже об их совокупности, что целостное их восприятие окажется не по плечу и наиболее сильным учащимся!

Завершим нашу статью небольшой подборкой задач, относящихся к рассматриваемому классу. (Не будем повторять здесь те задачи, формулировки которых уже приводились в тексте статьи).

Нетрудно понять, что возможности пополнения этого списка практически неограничены. Приведенные здесь задачи следует рассматривать, не более, как примерные.

6. «Классическая» задача о бассейне (1-й вариант).

В бассейн проведены две трубы, через одну из которых, при выключенной второй, он наполняется за T_1 с, а через вторую, при выключенной первой, — опорожняется за T_2 с. (Форма и размеры бассейна — такие же, как и в задачах 4 и 5).

В момент времени $t=0$ открываются обе трубы, а значение уровня в этот момент составляет h_0 .

Используя программы, разработанные при решении задач 4 и 5, получить табличную

Окончание. Начало см.: Информатика и образование. 1992. № 3—4.

зависимость уровня воды h от времени t при следующих наборах исходных данных:

- а) $T_1=240$ с, $T_2=300$ с, $h_0=0.20$ м;
б) $T_1=150$ с, $T_2=300$ с, $h_0=0.50$ м.

7. «Классическая» задача о бассейне (2-й вариант).

В бассейн проведены две трубы, через одну из которых, при выключенной второй, он наполняется за T_1 с, а через вторую, при выключенной первой, — опорожняется за T_2 с. (Форма и размеры бассейна — такие же, как и в задачах 4 и 5).

В момент времени $t=0$ открываются обе трубы, а значение уровня в этот момент составляет h_0 .

Составить программу для получения табличной зависимости уровня воды h от времени t , при условии, что при $t=0$ открываются обе трубы, а начальное значение уровня составляет h_0 .

Выполнить программу для следующих наборов исходных данных:

- а) $T_1=240$ с, $T_2=300$ с;
б) $T_1=150$ с, $T_2=300$ с.

Примечание: имеется в виду программа, которая после ввода исходных данных всю последующую обработку ведет, не используя средства диалога с пользователем (кроме запроса о необходимости вывода очередного фрагмента итоговой таблицы).

8. «Катапульта-1» (время, дальность и высота).

С помощью катапульты, установленной на горизонтальной поверхности, телу массы m придается скорость V_0 , направленная под углом A к этой поверхности. Спротивление воздуха пропорционально квадрату скорости. Коэффициент пропорциональности Q зависит от формы и размеров тела и от характера его поверхности.

Составить программу, позволяющую находить время и дальность полета, а также — максимальную высоту.

Выполнить программу для следующих наборов исходных данных:

- а) $V_0=50$ м/с, $m=100$ кг, $Q=1$ кг/м, $A=20^\circ$;
б) то же, но $A=70^\circ$;
в) то же, но $A=45^\circ$;
г) то же, но $A=37^\circ$;
д) $V_0=100$ м/сек, $m=0.5$ кг, $Q=0.01$ кг/м, $A=40^\circ$.

9. «Катапульта-2» (таблицы).

Применительно к условиям задачи 8 составить программу для получения табличной зависимости от времени следующих величин: удаления тела от точки вылета по горизонтали;

его высоты над исходным уровнем; горизонтальной и вертикальной составляющих скорости.

Выполнить программу для каждого из на-

боров данных, указанных в задаче 8.

Примечание. Удобно обеспечить вывод на экран фрагментов таблицы так, чтобы переход к следующему фрагменту инициировался пользователем.

10. «Катапульта-3» (прицельная стрельба — 1).

Для катапульты, описанной в задаче 8, при заданных значениях A , m и Q требуется найти (с точностью ϵ) такое значение V_0 чтобы дальность полета оказалась равной заданной величине S .

Решение выполнить для следующих наборов исходных данных:

- а) $m=100$ кг, $Q=1$ кг/м, $S=70$ м, $\epsilon=0.01$ м/с, $A=40^\circ$;
б) то же, но $A=10^\circ$;
в) то же, но $A=80^\circ$.

Провести это решение двумя способами: используя программу задачи 8, и — с помощью новой программы, позволяющей получить решение за один прогон, и не требующей иного управления от пользователя, кроме ввода исходных данных.

Указание: Удобно предусмотреть в этой программе задание интервала (V_{\min} , V_{\max}) для поиска V_0 , а сам этот поиск проводить методом последовательного половинного деления интервала.

11. «Катапульта-4» (прицельная стрельба — 2).

Для катапульты, описанной в задаче 8, при заданных значениях V_0 , m и Q требуется найти (с точностью ϵ) такое значение A , чтобы дальность полета оказалась равной заданной величине S .

Решение выполнить для следующих наборов исходных данных:

- а) $m=10$ кг, $Q=1$ кг/м, $V_0=50$ м/с, $\epsilon=0.01^\circ$, $S=50$ м;
б) то же, но $S=80$ м;
в) то же, но $S=100$ м;
г) $m=100$ кг, $Q=1$ кг/м, $V_0=100$ м/с, $\epsilon=0.01^\circ$, $S=100$ м.

Провести это решение двумя способами: используя программу задачи 8, и — с помощью новой программы, позволяющей получить решение за один прогон, и не требующей иного управления от пользователя, кроме ввода исходных данных.

Указание: учесть, что в зависимости от исходных данных могут найтись или два, или один, или — ни одного угла, удовлетворяющего условию задачи.

12. «Катапульта-5» (максимальная дальность).

Для катапульты, описанной в задаче 8, при заданных значениях V_0 , m и Q требуется найти (с точностью ϵ) такое значение A , чтобы дальность полета оказалась максимальной, а также — саму эту максимальную дальность.

Решение выполнить для следующих наборов исходных данных:

- $m=100$ кг, $Q=1$ кг/м, $V_0=20$ м/с; $e=0.01^\circ$,
- то же, но $V_0=50$ м/с;
- то же, но $V_0=100$ м/с;
- $m=100$ кг, $Q=1$ кг/м, $V_0=400$ м/с. $e=0.01^\circ$,

Провести это решение двумя способами: используя программу задачи 8, и — с помощью новой программы, позволяющей получить решение за один прогон, и не требующей иного управления от пользователя, кроме ввода исходных данных.

13. «Катапульта-6» (экстраполяция).

Для катапульти, описанной в задаче 8, при известных величинах угла вылета A_1 и начальной скорости V_01 дальность полета оказывается равной S_1 . Требуется найти такое значение начальной скорости V_02 , чтобы при величине угла вылета A_2 дальность полета была равной S_2 .

Выполнить программу для следующих наборов исходных данных:

- $A_1=20^\circ$, $V_01=40$ м/с, $S_1=80$ м, $A_2=40^\circ$, $S_2=100$ м;
- $A_1=35^\circ$, $V_01=100$ м/с, $S_1=150$ м, $A_2=35^\circ$, $S_2=200$ м;
- $A_1=35^\circ$, $V_01=100$ м/с, $S_1=150$ м, $A_2=35^\circ$, $S_2=300$ м.

14. «Упругие колебания — 1» (таблица).

Вдоль вертикально подвешенной пружины проведена координатная ось так, что нижний конец пружины оказывается в положении $x=0$. Сама ось направлена сверху вниз. Жесткость пружины равна k Н/м.

К пружине подвешивают груз массой m кг и без сообщения начальной скорости отпускают.

Составить программу, позволяющую получить табличную зависимость координаты точки подвеса груза x от времени t .

Выполнить программу для следующих наборов исходных данных:

- $k=60$ (Н/м), $m=15$ кг;
- $k=600$ (Н/м), $m=15$ кг;
- $k=600$ (Н/м), $m=150$ кг.

15. «Упругие колебания — 2» (график).

Для пружины, описанной в задаче 14, составить программу построения графика зависимости x от t .

Выполнить эту программу для тех же наборов данных, что и в задаче 14.

16. «Затухающие колебания — 1» (таблица).

Движение пружины, описанной в задаче 14, на самом деле происходит с энергетическими потерями. Приближенно можно считать, что этому движению препятствует сила сопротивления, прямо пропорциональная скорости движения ее незакрепленного конца.

Составить программу, позволяющую полу-

чить табличную зависимость координаты этого конца пружины от времени.

Выполнить ее для следующих наборов исходных данных:

- $k=60$ Н/м, $m=15$ кг, $R=1$ л/с;
- $k=60$ Н/м, $m=15$ кг, $R=4$ л/с;
- $k=60$ Н/м, $m=15$ кг, $R=10$ л/с.

17. «Затухающие колебания — 2» (график).

Для пружины, описанной в задаче 16, составить программу построения графика зависимости x от t .

Выполнить эту программу для тех же наборов данных, что и в задаче 16.

18. «Электроцепь с сопротивлением и самоиндукцией».

В электрической цепи поддерживается напряжение E . Сопротивление цепи равно R , коэффициент самоиндукции равен L .

В момент времени $t=0$ происходит замыкание цепи.

Составить программу для получения табличной зависимости силы тока от времени t .

Выполнить программу для следующих наборов исходных данных:

- $E=300$ В, $P=150$ ом, $L=30$ гн;
- $E=300$ В, $P=15$ ом, $L=300$ гн.

19. «Длина кривой».

Заменяя дугу кривой $y=f(x)$, заключенную между точками с абсциссами a и b вписанной в нее ломаной линией, и суммируя длины ее звеньев, мы получим приближенное значение длины этой дуги.

Составить программу, которая реализовывала бы эти вычисления.

Выполнить ее для следующих исходных данных:

- $y=x^2$, $a=0$, $b=1$ (дуга параболы);
- $y=\sqrt{1-x^2}$, $a=-1$, $b=1$ (полуокружность радиуса 1), (для случая b сравните полученный результат с точным значением, которое равно числу «пи»).

20. «Изолированный корень уравнения».

Если непрерывная функция $y=f(x)$ в промежутке $a < x < b$ имеет единственный корень, а значения функции в точках a и b противоположны друг другу по знаку, то приближенно найти значение этого корня можно методом последовательного половинного деления отрезка, на котором этот корень находится. Подсчитав значение $f(c)$, где $c=(a+b)/2$ — середина отрезка $[a, b]$, мы из двух образовавшихся его частей: $[a, c]$ и $[c, b]$ возьмем ту, на границах которой знаки $f(x)$ различны между собой. (Это различие знаков и говорит о том, что именно на этой части исходного отрезка располагается искомый корень). Далее с полученным отрезком повторяется такая же операция, и этот процесс продолжается до тех пор, пока разность между концами очередного от-

резка не окажется достаточно малой. Тогда середину этого отрезка и принимают за иско- мое значение.

Разумеется, если на каком-то шаге окажет- ся, что $f(c)=0$, то тем самым корень будет найден, и необходимость продолжать деле- ние отрезка отпадет.

Составить программу, реализующую эти вычисления.

Использовать ее для решения следующих уравнений:

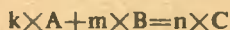
- а) $4 \times x^3 - x^2 - x - 5 = 0$ на отрезке [1; 2];
- б) $2 \times \sin(x) - x = 0$ на отрезке [1.57; 3.14];
- в) $10^x - x - 2 = 0$ (отрезок, содержащий корень, найти самостоятельно).

21. «Скорость химической реакции».

Средней скоростью химической реакции образования некоторого вещества за проме- жуток времени от t_1 до t_2 называется отно- шение Dq/Dt , где $Dt=t_2-t_1$, а Dq — ко- личество (в молях) этого вещества в едини- це объема, образовавшееся за этот времен- ной интервал.

Аналогично определяется скорость реак- ции по уничтожению вещества. Скоростью реакции в момент времени t называется пред- ел средней ее скорости за промежуток от t до $t+Dt$ при Dt , стремящемся к 0.

В очистном сооружении вредное вещество А, вступая в реакцию с нейтрализатором В, дает безвредное вещество С, выпадающее в осадок:



Целочисленные коэффициенты k , m и n указывают количество молекул каждого из веществ, участвующих в элементарной ре- акции.

Начальная концентрация вещества А со- ставляет P (мол/м³), вещества В — q (мол/м³): а скорость уничтожения А вы- ражается формулой:

$$W = S \times p \times q$$

где S — постоянная величина, а p и q , со- ответственно, — концентрации веществ А и В.

Составить программу для получения таб- личной зависимости p и q от времени t .

Выполнить эту программу для следующих исходных данных:

- а) $S=0.000001$, $k=1$, $m=3$; при $t=0$: $p=1000$, $q=1000$;
- б) $S=0.000001$, $k=3$, $m=1$; при $t=0$: $p=1000$, $q=1000$;
- в) $S=0.000001$, $k=1$, $m=3$; при $t=0$: $p=1000$, $q=10\ 000$.

22. «Удавы и кролики — 1» (простейший расчет).

Модель «хищник-жертва» (по В. Вольтер- ра) имеет вид:

$$Dq = (B - C \times H) \times G \times Dt$$

$$Dh = (A \times G - D) \times H \times Dt$$

где G — численность «жертв» (кроликов), H — численность «хищников» (удадов), Dq — приращение G , а Dh — приращение H за «элементарный» промежуток времени Dt . A — «коэффициент размножения» удадов, B — «коэффициент размножения» кроликов, C — «коэффициент поедания» кроликов уда- вами, а D — «коэффициент смертности» удадов.

Если G и/или H станут меньше своих «критических значений» $G(kp)$ и $H(kp)$, то процесс прекращается (популяция погибла).

Составить программу, позволяющую полу- чить табличную зависимость G и H от вре- мени t .

Выполнить эту программу для следующих исходных данных:

а) $A=0.001$, $B=5$, $C=0.002$, $D=10$, $G(kp)=200$, $H(kp)=200$, начальные значе- ния $G_0=15\ 000$, $H_0=2000$.

б) то же, но $G_0=2000$, $H_0=15\ 000$.

Для каждого из этих пунктов счет по программе провести при $Dt=0.002$, N (число шагов расчета перед выводом строки) = 50 и для $Dt=0.01$, $N=10$. Чем объяснить не- значительные расхождения в полученных ре- зультатах?

23. «Удавы и кролики — 2» (уточненный расчет).

Применительно к условию задачи 22, для достижения большей точности вычислений (без уменьшения Dt и, значит, соответствующего увеличения времени счета) при вы- числении приращений Dq и Dh использовать для каждого элементарного промежутка вре- мени Dt формулы Вольтерра дважды: сна- чала найти по ним Dq и Dh , затем — «про- межточные» значения G_1 и H_1 :

$$G_1 = G + Dq/2, \quad H_1 = H + Dh/2,$$

которые будут приближенно соответствовать середине элементарного временного проме- жутка, а затем — вновь обратиться к фор- мулам Вольтерра, подставив в их правой части G_1 и H_1 вместо G и H .

Выполнить модифицированную программу для тех же исходных данных, что и в зада- че 22, и сравнить результаты.

24. Удавы и кролики» (графики).

В условиях задачи 23 составить про- грамму, которая выводила бы на экран графи- ки зависимости G и H от t .

Выполнить эту программу при тех же ис- ходных данных, что и в задаче 22.

И. ХОРОШЕВА

Программа курса «Информатика»

(для VIII—IX классов школ, принимающих участие в проекте «Пилотные школы»)

Основная цель описываемого в статье курса «Информатика» состоит в том, чтобы сформировать у учащихся представление о современных информационных технологиях и элементарные умения практического использования наиболее эффективных и широко распространенных из них, ознакомить учащихся с ролью информационных технологий в современном обществе, их влиянием на состояние и развитие науки, техники и технологии, на развитие и распространение самих информационных технологий.

Другая важная цель — подготовка учащихся к активному и эффективному использованию информационных технологий при изучении других дисциплин.

Основные задачи курса «Информатика»: ознакомление с основными устройствами современного компьютера, овладение навыками практической работы с персональным компьютером, овладение основными умениями алгоритмизации и планирования решения сложных задач, формирование представлений о современной компьютерной графике и ее применении в различных сферах человеческой деятельности, ознакомление с основными понятиями и методами практического использования одного из графических пакетов, обязательно включающего элементы компьютерной анимации, формирование представлений о современных интегрированных системах и их использовании в различных сферах человеческой деятельности, ознакомление с основными понятиями и методами решения задач в одной из интегрированных систем, формирование представлений об этапах решения задач с использованием компьютера, математической и компьютерной модели, о методах и целях компьютерного моделирования, знакомство с основными понятиями и методами практического использования адаптированной версии одного из пакетов компьютерного моделирования, а также с компьютерными моделями некоторых объектов и систем, ознакомление с современными системами телекоммуникаций, формирование представлений об информационных технологиях как

системе технических и программных средств и методах их использования, о месте информационных технологий в жизни современного общества и о перспективах развития информационных технологий.

Курс «Информатика» не только вооружает учащегося новым инструментарием для обработки информации, но и дает ему систему методов использования этого инструментария для изучения и анализа явлений, объектов и систем.

Естественно, что такой курс не может вводиться изолированно, он должен быть поддержан изменениями в содержании, форме и методах преподавания других дисциплин. Такие изменения необходимы для того, чтобы отразить то влияние, которое оказывают информационные технологии на современную науку, технику, технологию и все сферы общественной жизни.

Курс организован как взаимосвязанная система 7 тем (2 ч в неделю, всего 136 ч).

Тема I. Знакомство с компьютером (10 ч)

При изучении темы «Знакомство с компьютером» учащиеся должны получить представление о предмете информатики, об информационных технологиях, об изменениях, произошедших в жизни общества после «компьютерной революции». При изучении этой темы учащиеся должны получить представление об устройстве и возможностях компьютера, а также освоить навыки, необходимые для практической работы с компьютером.

1. Введение (2 ч). Информация вокруг нас. Компьютер как инструмент для обработки информации. Роль компьютерных систем в современном мире.

2. Первое знакомство с компьютером (2 ч). Внешний вид и основные устройства компьютера. Управление работой программы при помощи манипулятора «мышь».

3. Знакомство с клавиатурой (2 ч). Алфавитно-цифровые клавиши и курсор. Редактирование текста. Прописные и строчные буквы. Правила работы с алфавитно-цифровой клавиатурой. Клавиатура русск и управляющих символов. Переключение с цисского шрифта на латинский и обратно. Функциональные клавиши.

4. Компоненты компьютера (4 ч). Основные этапы работы с информацией. Типы устройств, составляющих компьютер. Критерии сравнения различных устройств одной группы. Программное обеспечение компьютера.

После изучения темы I учащиеся должны иметь представление о том, что понимается под термином «информация» в различных областях человеческой деятельности, об информатике как науке, информационных технологиях, изменениях, произошедших в жизни людей после «компьютерной революции», о назначении основных клавиш клавиатуры, правилах редактирования текста с помощью клавиш редактирования, основных режимах работы клавиатуры, об основных компонентах компьютера, их типах и выполняемых ими функциях.

Учащиеся должны уметь различать и называть основные устройства компьютера: системный блок, монитор, клавиатуру и манипулятор «мышь», выбирать из меню при помощи «мыши», определять местонахождение курсора на экране и изменять положение курсора при помощи клавиш управления курсором, переключать режимы работы клавиатуры, вводить и редактировать текст, определять, к какой группе устройств относится конкретное периферийное устройство компьютера, определять, какой набор компонентов компьютера требуется для решения определенной задачи.

Тема II. Основы алгоритмизации (24 ч)

Изучение темы «Основы алгоритмизации» ориентировано на привитие учащимся навыков и умений, необходимых для решения задач, требующих продуманной последовательности действий, анализа содержания и структуры исходных данных. Для достижения поставленных целей акцент делается не на запоминании конструкций какого-либо языка программирования, а на практической работе в стиле игры с удобными, понятными и привлекательными для школьников исполнителями. Навыки и умения, полученные при изучении основ алгоритмизации, активно используются и развиваются при изучении последующих тем и разделов курса.

1. Основные понятия (2 ч). Понятие об исполнителе и системе его команд. Примеры исполнителей. Понятие алгоритма. Запись алгоритмов достижения цели для простого исполнителя. Линейные алгоритмы. Решение задач.

2. Процедуры (4 ч). «Встроенные» команды исполнителя. Дополнительные команды исполнителя как метод упрощения записи алгоритма. Понятие процедуры. Определение дополнительных команд простого исполнителя с помощью процедур. Понятие об основных этапах работы с простым исполнителем. Решение задач.

3. Циклы (4 ч). Понятие о типах алгоритмов. Команда повторения. Запись алгоритмов для про-

стого исполнителя с использованием команд повторения. Решение задач.

4. Ветвление (4 ч). Понятие о зависимости состава и последовательности действий исполнителя при решении задачи от исходных данных. Команда ветвления. Запись алгоритмов для простого исполнителя с использованием команды ветвления. Решение задач.

5. Данные (8 ч). Понятие о данных как об объектах действий исполнителя. Примеры данных. Понятие об организации (структуре) данных. Структуры таблиц и стек. Простой исполнитель, ориентированный на работу с простыми структурами данных. Решение задач.

6. Заключение (2 ч). Разнообразие исполнителей. Примеры различных исполнителей. Практическое решение задач с использованием различных исполнителей.

После изучения темы II учащиеся должны иметь представление об исполнителе и системе его команд, алгоритме как организованной последовательности команд, обеспечивающей достижение цели, поставленной перед исполнителем; о требованиях, предъявляемых при записи алгоритмов для конкретных исполнителей; о процедуре как способе определения дополнительных команд исполнителя; об основных типах алгоритмов (линейном, циклическом, ветвящемся); о данных и способах их организации.

Учащиеся должны уметь составлять и записывать алгоритмы для достижения цели простых исполнителей с использованием основных типов алгоритмических структур; составлять и записывать процедуры для простых исполнителей, практически работать с простыми исполнителями, имитируемыми на ПЭВМ.

Тема III. Элементы компьютерной графики (22 ч)

Тема «Элементы компьютерной графики» дает учащимся представление об одном из ведущих направлений современных информационных технологий. Учащиеся овладевают навыками графического диалога с компьютером, учатся формировать изображение на экране компьютера. При подготовке компьютерных фильмов с элементами анимации в значительной мере развиваются навыки и умения, полученные при изучении темы «Основы алгоритмизации».

1. Основные понятия (1 ч). Системы компьютерной графики, их назначение и сфера применения. Примеры использования систем компьютерной графики. Состав и назначение простой графической системы.

2. Графический редактор (9 ч). Графические примитивы и их использование для построения графических изображений. Средства для цветного исполнения графических изображений и их использование. Средства организации и использования библиотеки графических изображений. Средства коррекции графических изображений. Практическая работа с графическим редактором простой графической системы.

3. Редактор сценариев (12 ч). Понятие о компьютерной мультипликации. Понятие о сценарии компьютерного мультфильма как об алгоритме смены графических изображений. Команды для составления сценария. Параметры команд сценария компьютерного мультфильма и форма их представления. «Прокрутка» компьютерного мультфильма.

После изучения темы III учащиеся должны иметь представление о современных системах компьютерной графики, их назначении, сферах применения и возможностях; составе типичной графической системы, назначении и функциях графического редактора, назначении и функциях редактора сценариев, назначении и функциях модуля просмотра сценариев.

Учащиеся должны уметь практически работать с системой меню простой графической системы, пользоваться графическим меню графического редактора для выбора инструментов, линий, цветов и узоров; использовать сервис меню для работы с библиотекой слайдов, корректировки изображения, сохранения и загрузки файлов; конструировать несложное изображение на компьютерном «холсте», загружать и сохранять как весь сценарий компьютерного мультфильма, так и его части; просматривать сценарий полностью или частично, конструировать простой собственный сценарий.

Тема IV. Деловые применения ЭВМ (34 ч)

Основная цель изучения темы «Деловые применения ЭВМ» — познакомить учащихся с одним из интегрированных пакетов, дать представление о возможностях применения компьютера с использованием пакетов такого рода, в частности в процессе обучения в общеобразовательной школе.

1. Введение. Основные понятия (2 ч). Деловые применения ЭВМ как наиболее распространенная форма практического использования компьютеров. Состав, назначение и интерфейс интегрированной системы для деловых применений.

2. Электронные таблицы (6 ч). Знакомство с электронной таблицей. Элементы электронной таблицы: строка, столбец, ячейка, ссылка, группа. Описание элементов электронной таблицы. Формулы, их назначение, запись и редактирование. Некоторые встроенные функции системы Фреймонтаж. Операции со строками, столбцами, группами. Форматы отображения чисел. Создание и заполнение таблицы. Решение практических задач.

3. Деловая графика (4 ч). Назначение деловой графики. Основные типы графиков. Иллюстрирование содержимого электронной таблицы с помощью диаграмм. Основные параметры, определяющие диаграмму. Автоматическое построение графиков. Решение практических задач.

4. Базы данных (10 ч). Понятия базы данных и системы управления базами данных. Назначение баз данных. Табличная форма представления базы данных. Понятие записи и поля. Другие формы представления базы данных: стандартная

и в виде картотеки. Просмотр базы данных. Операции поиска по значению поля. Использование «слепых символов» для записи шаблона поиска. Ограничения поиска по значению поля. Понятие фильтрации базы данных. Формула фильтрации. Понятие сортировки. Виды сортировки. Сочетание сортировки и фильтрации.

Построение диаграмм, иллюстрирующих содержание базы данных. Создание и редактирование базы данных. Решение практических задач.

5. Текстовый редактор (12 ч). Понятие текстового редактора. Назначение текстового редактора. Редактирование текста. Операции форматирования и стилизации текста. Понятие составного каркаса. Подготовка документов с использованием составных каркасов. Понятие типового письма. Создание шаблона письма. Автоматическое заполнение шаблонной информацией из базы данных.

Использование меню «Печать». Печать с использованием принтера и в файл печати.

После изучения темы IV учащиеся должны иметь представление об интегрированном пакете как о целостной системе, предназначенной для решения задач; основных понятиях, связанных с интегрированным пакетом «Фреймонтаж»: стеллаже, документе, каркасе; о системе меню интегрированного пакета; об электронной таблице; основных понятиях, связанных с электронной таблицей: ячейке, строке, столбце, группе, ссылке; о формулах, форматах представления и точности записи чисел, базах данных и системах управления базами данных, об основных понятиях, связанных с базами данных: записи, поле, форме представления; основных операциях, реализуемых при работе с базами данных: поиске, сортировке, фильтрации; о деловой графике, реализуемой интегрированным пакетом; текстовом редакторе, об основных понятиях, связанных с текстовым редактором: текст, слово, абзац, страница; основных операциях при работе с текстовым редактором: редактировании, форматировании и стилизации текста.

Учащиеся должны уметь выполнять основные операции с документами (загрузка, сохранение, удаление); читать, записывать, копировать и редактировать формулы; создавать новые и удалять строки и столбцы электронной таблицы записи и поля базы данных; копировать и переносить фрагменты электронной таблицы и базы данных; создавать новые таблицы, базы данных и текстовые файлы; создавать составные документы, строить диаграммы, иллюстрирующие данные, содержащиеся в таблицах и базах данных; изменять форму представления базы данных; осуществлять поиск в базе данных, записывать и использовать формулы фильтрации базы данных; осуществлять сортировку базы данных по значению одного или нескольких полей, создавать электронные таблицы-отчеты по информации из базы данных, редактировать, форматировать и стили-

зывать текст; заполнять типовые письма по информации из базы данных.

Тема V. Элементы компьютерного моделирования (38 ч)

Изучая тему «Элементы компьютерного моделирования», учащиеся получают общее представление о методе моделирования, о моделировании с помощью компьютера, получают начальные навыки применения компьютерного моделирования для решения прикладных задач.

1. Основные понятия компьютерного моделирования (14 ч). Понятие компьютерной модели. Типы моделей. Описание элементов модели. Наблюдение и анализ поведения модели. Понятие адекватности модели и границ ее применения.

2. Технология компьютерного моделирования (14 ч). Специализированная среда компьютерного моделирования. Представление информации о моделируемых объектах и средства ее отображения. Построение моделей в специализированной среде компьютерного моделирования.

3. Применение компьютерных моделей (10 ч). Примеры задач, решаемых с помощью компьютера на основе моделирования.

После изучения темы V учащиеся должны иметь представление о типах моделей, специфике компьютерного моделирования, об информационных моделях, о технологии моделирования, современных компьютерных средствах построения моделей.

В. ЛЕЩИНЕР, Н. МАТВЕЙКИНА

Использование интегрированных пакетов

В средних учебных заведениях в связи с проектом «Пилотные школы» появились персональные ЭВМ IBM PS/2, которые гораздо мощнее любого из компьютеров, использовавшихся ранее в учебном процессе. Для компьютеров этого типа существует большое разнообразие программных средств, рассчитанных на начинающего или недостаточно квалифицированного пользователя.

Традиционно школьный курс информатики был ориентирован в первую очередь на обучение детей программированию. Концепция, принятая в проекте «Пилотные школы», состоит в том, что в базовом курсе информатики вообще не должно быть обучения основам программирования. Подавляющее большинство современных пользователей не прибегают к программированию: существует значительное количество прикладных пакетов, ориентированных на решение как достаточно узких, так и очень широких классов задач, поэтому не требуется составлять свою программу — можно просто выбрать

Учащиеся должны уметь строить простейшие модели, применять готовые компьютерные модели для решения задач, применять специализированные моделирующие программы.

Тема VI. Компьютеры в системах передачи информации (6 ч)

Тема «Компьютеры в системах передачи информации» знакомит учащихся с современными системами передачи информации и их ролью в жизни общества.

Значение систем передачи информации для жизни современного общества. Место компьютера в системах передачи информации. Компьютер как средство общения. Примеры использования компьютеров в системах передачи информации.

Тема VII. Компьютеры вчера, сегодня и завтра (2 ч)

Заключительная тема курса «Компьютеры вчера, сегодня и завтра» позволяет учащимся определить место информационных технологий в жизни современного общества и формирует представление о перспективах развития информационных технологий.

Поколения ЭВМ. Совершенствование программного обеспечения и средств его разработки. Новые информационные технологии в современном обществе. Перспективы развития информационных технологий.

наиболее подходящую из уже существующих.

Базовый курс информатики, в нашем понимании, должен прежде всего познакомить школьников с наиболее распространенными типами программного обеспечения персональных компьютеров и научить их работать с некоторыми программами. Ясно, что в этом случае нельзя обойти электронные таблицы и системы управления базами данных, текстовые процессоры и программы для телекоммуникаций. В каждой из этих областей существуют программы-лидеры, наиболее распространенные в мире и прочно ассоциирующиеся в сознании пользователей с данным типом программного обеспечения.

Конечно, можно было бы ориентировать курс на развитие навыков работы именно с такими средствами. Мы, однако, решили пойти по другому пути: зачем тратить время и силы на формирование у школьников навыков работы с разнотипным программным обеспечением, если нельзя точно знать, с какими конкретными программами им при-

дется работать? Жизнь не стоит на месте, и к тому времени, когда наши выпускники начнут использовать компьютер в практической работе, появятся другие средства и сформированные в школе навыки работы могут остаться невостребованными. В то же время хотя различные программные средства одного направления (электронные таблицы, например) и имеют различный интерфейс, по-разному управляются, общие принципы их использования остаются неизменными. Так же как научившийся хорошо программировать на Паскале школьник впоследствии без труда освоит программирование на языке Си, так же и освоивший одну программу электронных таблиц без особого труда научится управлять другой. Эти рассуждения привели нас к выводу о том, что в целях экономии времени и сил целесообразнее вместо разнородного программного обеспечения курса информатики использовать интегрированный пакет программ. Пусть каждая из составляющих этого пакета будет слабее наиболее распространенных программ своего типа — не беда, школьник вряд ли использует все возможности даже интегрированной системы, зато эти программы будут иметь единый интерфейс, соответственно эффективность применения их в учебном процессе резко повысится.

Интегрированные пакеты имеют еще одно преимущество, едва ли не решающее в данном случае. Это возможность интеграции программ и данных. С нашей точки зрения, нельзя не показать школьнику, что компьютер способен рассылать стандартные письма, выбирая адреса и другую информацию из базы данных, построить диаграмму на основании содержания базы данных, подсчитав значения некоторых величин с помощью электронных таблиц, и т. д. Без этого представление школьников о компьютере как средстве автоматизации умственного труда будет неполным. Таким образом, мы пришли к выводу о необходимости включения в базовый курс информатики для VIII—IX классов «Пилотных школ» темы «Деловые применения ЭВМ», на которую отводится 34 часа — больше четверти всего курса.

Помимо всего прочего, мы прекрасно понимали, что появление в школе такого мощного средства автоматизации канцелярского труда отразится и на работе школьной администрации, поэтому хотелось, чтобы выбранная система давала возможность создавать самостоятельные приложения, направленные на механизацию наиболее распространенных в школе операций, связанных с делопроизводством. Далее мы стали выбирать наиболее подходящий для этой цели пакет. Нужна была система, во-первых, не

слишком дорогая и не слишком мощная, учитывая финансовое положение органов образования и технические характеристики имеющихся в школах компьютеров (большинство из них имеет ОЗУ на 1 Мбайт, НГМД на 1.44 Мбайт, 286 процессор, но не имеет жесткого диска), и, во-вторых, достаточно простая в управлении, по возможности с более дружелюбным интерфейсом. Кроме того, мы хотели иметь полностью русифицированную и допускающую создание национальных версий систему.

В нашем распоряжении таких систем оказалось три — это достаточно широко распространенные пакеты: Works фирмы Microsoft, Open Access и, наконец, система Framework II фирмы Ashton-Tate.

Каждый из этих пакетов обладает какими-то безусловными преимуществами. У системы Framework это прежде всего возможность работы со многими окнами на одном экране. У Works — доступный интерфейс, возможность работать с «мышью» и простота управления. У Open Access — очень мощная СУБД и большие возможности деловой графики. Естественно, что достоинства систем этим не исчерпываются.

Однако применительно к задачам первоначального обучения работе с компьютером преимущество пакета Open Access обернулось недостатком: создание даже простенькой базы данных в этом пакете обязательно требует описания характеристик всех полей, что сразу же осложняет процесс обучения. Красивые диаграммы можно построить только для данных из электронных таблиц, поэтому, чтобы построить диаграмму на основании значений из базы данных, требуется сначала экспортировать данные в электронную таблицу, что тоже не совсем просто. Пакет, заслуживший достаточно широкую популярность в деловом мире, для решения наших задач оказался непригоден.

Пакет MS Works (версия 2.0), любезно предоставленный в наше распоряжение московским представительством фирмы Microsoft, привлек нас своей простотой и доступностью. Он легко, вместе со всеми файлами приложений, умещается на дискете, с ним можно работать в сети (PC LAN, Base Band), однако файлы на учебных компьютерах при работе в сети можно было открывать «только для чтения». Пакет полностью русифицирован фирмой Microsoft, в том числе получили русские названия арифметические функции в электронных таблицах, что достаточно нетрадиционно. При русификации несколько изменилась архитектура пакета, перегруппировались меню и исчезли некоторые режимы. В частности, исчез режим создания и редактирования формата записи базы данных, он оказался объединенным с

режимом заполнения и редактирования записей. Это слегка затрудняет обучение школьников стадиям работы с базой данных. После русификации появился другой пакет, имеющий безусловное родство с оригинальной системой, но не совпадающий с ней полностью.

Ученики овладевали работой с Works достаточно быстро. Без затруднений были освоены электронные таблицы, прежде всего благодаря хорошо построенной системе меню, в большинстве случаев позволяющей вместо ввода символов ограничиться простым указанием на экране или выбором из списка ограниченных возможностей. (Framework II, помимо того, что в нем нельзя указывать «мышью», требует от ученика интенсивной работы с клавиатурой для ввода содержательной информации.) Очень большой энтузиазм ребят, как и ожидалось, вызывает работа с диаграммами. Works позволяет легко менять внешний вид уже построенной диаграммы, выбирать цвет, подписывать оси и т. д. Главное ограничение этого режима заключается в невозможности показать на экране одновременно таблицу и диаграмму. Хотя пакет позволяет работать с несколькими окнами, все они должны быть в текстовом, а не графическом режиме. Это требует от учащихся хорошо помнить содержание таблицы при работе с диаграммами. (Естественно, это является ограничением только в контексте задач обучения — нельзя, например, показать на экране: вот число, являющееся значением данной ячейки электронной таблицы и вот столбик/сектор диаграммы, представляющий это число.)

Раздел, посвященный работе с базами данных, осваивался с гораздо большим трудом. Одной из главных проблем было то, что система не дает ограничивать поле для записи, оно представляется очень большим и очень нечетко структурируется на страницы. В отличие от других пакетов на экране зрительно не разделяются названия полей базы данных и их содержание. В силу этого учащиеся далеко не всегда могли сказать, какой фрагмент поля представлен на экране: многие писали на экране лишние строки, которые потом не были видны, но влияли на процессы поиска и сортировки.

Кстати, электронные таблицы в Works организованы таким же образом, их размер всегда постоянен. Методически это, пожалуй, недостаток, так как привычка определять размер таблицы при ее проектировании представляется нам очень существенной. При работе с пакетом Framework II этот навык вырабатывается более естественно, хотя и эта система позволяет определять размеры таблиц и баз данных по умолчанию. Система Works позволяет представлять базы

данных как в анкетной форме, так и в табличной, при этом базовой формой является анкетная: именно она появляется на экране при создании новой базы данных. (Система Framework формирует таблицу.)

Неудобной для работы с русской версией пакета оказалась одна, но весьма существенная деталь процесса формирования отчета по базе данных: при переходе в режим создания отчета на экране появляется готовый отчет, представляющий полную таблицу по всей базе данных. Пользователю приходится стирать лишние описатели столбцов перед работой. Было бы более правильным начинать работу по проектированию отчета с чистого бланка — как это сделано в англоязычном варианте системы.

Текстовый процессор системы Works близок к оптимальному для школьного курса. Он позволяет использовать различные виды шрифтов (на экране выделяются цветом), большой набор специальных символов, содержит широкий сервис для редактирования.



Очень удачно организована в Works работа с абзацами: в меню содержатся как отдельные возможности по форматированию абзаца (расположение текста, абзацный отступ, выравнивание и т. д.), так и режим комплексного описания всех характеристик абзаца. Система легко перестраивается с измерений в дюймах на измерение в сантиметрах или символах. В полном объеме осуществляются в текстовом процессоре Works контекстный поиск и замена групп символов.

Интеграция программ пакета Works реализована в достаточно полном объеме. Существует специальный режим заполнения стандартных писем информацией из базы данных. В подготовленный в текстовом процессоре текст можно вставить в виде фрагмента диаграмму, при этом специальная функция предварительного просмотра позволит перед печатью увидеть расположение текста и рисунков на странице. Суммируя, можно сделать вывод, что пакет MS Works

в целом подходит для задач первоначально-го обучения работе с базовым программным обеспечением. Простота и доступность в сочетании с внешней привлекательностью делают его очень перспективным с точки зрения использования в средней школе. Однако система содержит и некоторые вполне преодолимые недостатки, большая часть которых была отмечена выше. Одним из недостатков является отсутствие языка программирования и в силу этого невозможность создания специализированных приложений в среде этого пакета для решения задач автоматизации школьного делопроизводства.

В целом можно сделать вывод, что русский вариант системы Works фирмы Microsoft вполне подходит для задач обучения информатике в средней школе, если бы не его высокая цена, не доступная для большинства учебных заведений.

Возможность работы с несколькими окнами сразу (в том числе совмещение графиков и текста) и расположение их на экране по желанию пользователя — важнейшее достоинство системы Framework. В основе системы лежит развернутая метафора «рабочего стола». Соответственно объекты на экране носят метафорические названия: документы, стеллажи и пр. Это крайне упрощает задачу объяснения организации хранения информации в памяти компьютера: диски — это стеллажи, в которых имеются полки (директории), универсальные для системы клавиши <Внутри> и <Наружу> действуют в области стеллажей так же, как и в области документов. Единственный недостаток системы Framework II в части организации работы с файлами (преодоленный, кстати, системой Framework III) — невозможность фильтрации содержимого стеллажа, т. е. маскирования файлов по некоторому шаблону. В остальном идеология работы с дисками очень проста и объясняется метафорически говоря, на пальцах.

В системе Framework II очень мощные электронные таблицы. Существует также язык программирования FRED, программы на котором синтаксически представляют собой формулы, помещаемые в ячейки электронных таблиц. Огромным достоинством системы является то, что эти формулы также могут быть записаны и в записи базы данных, и даже в текстовые каркасы.

Ведущим для системы является понятие каркаса (фрейма) — простой или составной структуры, являющейся либо самостоятельным документом, либо составной частью какого-либо другого каркаса. Эта достаточно сложная для понимания идея графически реализуется максимально наглядно, чем объясняется сравнительная легкость ее усвоения учащимися. Framework II дидактически

очень эффективно поддерживает идею создания документа по частям: либо «сверху вниз» — от общего плана к его детализации, либо «снизу вверх» — создание отдельных фрагментов с последующим их объединением в единый документ. Более полной интеграции пакета, чем та, которая существует в системе Framework II, трудно себе представить.



Большой энтузиазм учащихся вызывает возможность построить на экране составной каркас, состоящий, например, из электронной таблицы и двух диаграмм. Самое эффективное — когда в результате изменения значения одной из ячеек таблицы автоматически перерисовываются диаграммы. Это не просто красивый трюк — это наглядная иллюстрация достаточно непростой, но фундаментальной идеи о разных формах представления информации.

В литературе можно встретить сетования по поводу сравнительной слабости СУБД пакета Framework II. С нашей точки зрения, для задач обучения эта слабость оборачивается преимуществом: внимание учащихся не отвлекается на детали, главные же операции: фильтрация и сортировка базы данных — выполняются без проблем. Синтаксис фильтрующей формулы позволяет акцентировать внимание учащихся на различии между именем переменной и ее содержанием (в системе Framework II фильтрующая формула записывается достаточно традиционно: <имя поля> <знак отношения> <значение поля> — причем возможно использование логических функций), необходимость записи фильтра в область формул всей базы данных провоцирует дополнительное обсуждение принципов организации баз данных, что совсем неплохо.

С точки зрения обучения у системы Framework II есть ряд недостатков. Главный среди них — невозможность работы с «мышью». Этот недостаток преодолен в системе Framework III. Оказалось, что и к системе Framework II можно подключить драйвер мыши. Среди других недостатков — довольно скромное использование графиче-

ских возможностей компьютера для организации интерфейса, нетрадиционное использование клавиши Insert для перехода в область меню, усложненная процедура работы с датами и т. д. Определенным недостатком является отсутствие режима формирования отчета по базе данных: для этой цели приходится связывать базу данных с электронной таблицей. Однако эти недостатки с лихвой восполняются многочисленными достоинствами пакета.

Для проекта «Пилотные школы» фирма Ashton-Tate подготовила специальную двуязычную (русско-английскую) школьную версию пакета Framework II, получившую название «Фреймонтаж». Разработка спецификации для системы «Фреймонтаж» велась с активным участием педагогов и методистов центра поддержки проекта «Пилотные школы» (А/О КУДИЦ), при этом были учтены как требования учебного процесса, так и особенности конфигурации комплекса средств вычислительной техники, поставляемого в рамках проекта. (В системе «Фреймонтаж» реализована возможность использования «мыши» для управления курсором.) Экспериментальные занятия с прототипом системы «Фреймонтаж» проходили в московской школе № 1217 в июне 1991 г. и даже для нас был удивителен тот энтузиазм, с которым учащиеся 13 лет во время летних каникул ходили на уроки. «Занятия дали нам ощущение уверенности в общении с компьютером», — говорили нам школьники, а в этом и состояла наша задача. В 1991/92 учебном году в значительной части «Пилотных школ» ученики получили возможность познакомиться с системой «Фреймонтаж».

Итак, интегрированные пакеты появились в программе курса информатики современ-

ной школы. Задача состоит в том, чтобы идти дальше: думать об использовании информационных технологий в преподавании других школьных предметов и повседневной школьной жизни. В настоящее время в учебно-методическом центре поддержки проекта «Пилотные школы» (А/О КУДИЦ) уже создаются такие системы: это пакет для автоматизации работы школьной канцелярии, базы данных по истории России, набор программ для проведения лабораторных работ по физике, комплекс программных средств по географии и многое другое. Система «Фреймонтаж» используется в них только как оболочка, внутри которой создаются специализированные приложения. Эти пакеты разрабатываются нами таким образом, чтобы пользоваться ими могли люди с разным уровнем подготовки: от учителей общеобразовательных предметов, имеющих только самые общие представления о компьютере, до познавших все тонкости программирования на языке FRED учащихся специализированных классов.

В этой статье не был затронут вопрос об использовании пакета Framework II для организации работы в области телекоммуникаций — у нас пока нет соответствующего опыта, однако есть определенные планы в этом направлении. Еще одну возможность пакета нам подсказал Брюс Маркварт, менеджер фирмы Ashton-Tate по странам Содружества и Восточной Европе: русификация системы выполнена настолько изящно, что для переключения языковой среды требуется нажатие всего лишь двух клавиш (Alt—F10) — в связи с этим у школьников появляются дополнительные возможности изучения английского языка. Время покажет, воспользуются ли они такой возможностью.

НЕМЕДЛЕННО ВЫШЛЕМ!

1. Методическое пособие «Учителю информатики» — это:
опыт работы передовых учителей, новаторов;
рекомендации методистов;
всевозможные варианты планирования учебного материала;
десятки задач по всем темам школьного курса;
методические рекомендации для работы с детьми V—VII классов.

Пособие «Учителю информатики» — это то, что вам нужно!

Стоимость пособия 273 рубля.

2. Задачник по основа информатики:
задачи по всем темам школьного курса;
решения к задачам на русском алгоритмическом языке и языке Бейсик.
Стоимость задачника 220 рублей.

Заявки присылайте по адресу: 683001, Петропавловск-Камчатский, Набережная, 26, ИУУ, «Ритм». Оплата наложенным платежом или по перечислению.

**Всероссийский институт повышения квалификации
инженерно-педагогических работников и специалистов
профтехобразования
предлагает**

1. Для преподавателей дисциплин «Основы информатики и вычислительной техники» и «Информатика и автоматизация производства» всех типов средних учебных заведений месячные курсы повышения квалификации в соответствии с действующими учебными программами.

2. Для преподавателей всех дисциплин, желающих использовать в учебном процессе новые информационные технологии обучения, двухнедельные курсы, ориентированные на преподавателей, не имеющих навыков работы с компьютером.

3. Для руководящих работников системы народного образования двух-трехнедельные курсы по подготовке к использованию персонального компьютера в управленческой деятельности.

4. Для специалистов, желающих получить или повысить производственный разряд по специальности «Оператор станков с ЧПУ», трехнедельные курсы на базе колледжа «Станкоэлектрон».

Обучение и проживание в общежитии для преподавателей и руководящих работников системы профтехобразования Российской Федерации (за исключением п. 4) бесплатное. Ориентировочная плата за обучение для других работников народного образования из расчета 500 рублей за месяц; для специалистов, указанных в п. 4, — 500 рублей за неделю.

Адрес: 192007, Санкт-Петербург, ул. Воронежская, д. 79.

Телефон для справок: (812) 166-67-20. Факс: (812) 166-34-34.

**Экспериментальный учебно-методический
исследовательский центр
информационных
технологий Миасского горно
предлагает**

**учебное-программное обеспечение
для ПЭВМ «Агат-7/9»:**

RobotLand — система поддержки курса информатики для младших школьников, аналог знаменитой «Роботландии», существовавшей ранее лишь для «Ямахи» и УКНЦ.

Робот + Измеритель + Чертежник — учебный исполнитель — система поддержки курса информатики по учебнику А. Г. Кушнйренко.

Пролог-Д — учебный интерпретатор Пролога — система поддержки курса информатики по учебнику В. А. Каймина.

GERA — универсальный учебный графический редактор, позволяющий работать со всеми графическими режимами «Агата».

Конструктор графиков тригонометрических функций — для компьютерных уроков математики.

RAM-диск — для отладки программ на Рапире.

Заявки с указанием типа «Агата» (7 или 9), типа дисководов (ЕС-5088/140К или ЕС-5323, МС-5311/840К) направлять по адресу: 456324, г. Миасс, Челябинская обл., а/я 140, ЭУМИЦИТ.

Телефон, факс: (8-35135) 2-14-75.



ТВ-ИНФОРМ-ОБРАЗОВАНИЕ



28
Современные технологии передачи и обработки информации открыли новые возможности перед системой образования. В свою очередь, образование, обслуживая потребности общества, предъявляет новые, высокие и непрерывно растущие требования к своему информационному обеспечению, к интенсивности и качеству обмена информацией, как между системой образования и обществом, так и внутри самой системы. Не вызывает сомнений тот факт, что существующие традиционные технологии информационного обмена не обеспечивают должный уровень оперативности, надежности и полноты передачи непрерывно возрастающего потока информации необходимой системе образования. В настоящее время не удастся на должном уровне решить даже относительно простые проблемы оперативного распространения административной информации от центральных органов управления к периферийным учреждениям. Учебные заведения и местные органы управления образованием испытывают острый дефицит в оперативной информации о современном состоянии педагогической науки, педагогических и методических инновациях, в учебно-методических материалах, педагогических программных средствах. Такой дефицит существенно снижает качество учебного процесса и вызывает значительные потери времени и средств, связанные с поиском, созданием, доставкой и многократным дублированием материалов. Международный опыт показывает, что создание глобальных информационных компьютерных сетей является неизбежным этапом на пути совершенствования системы образования и позволяет рассчитывать на существенный качественный скачок в ее развитии.

Интенсивное внедрение компьютеров в систему образования создает необходимую материально-техническую и интеллектуальную базу развития сетевых информационных систем. Увеличение стоимости традиционных методов доставки информации (почта, телефон, курьерская связь и т. п.) вместе с относительным снижением стоимости аппаратного и программного обеспечения новых информационных технологий

делают необходимым и экономически целесообразным развитие современных компьютерных информационных сетей. Новые средства коммуникации позволяют решить проблему оперативности, надежности и полноты передачи необходимого объема информации без катастрофически большого увеличения затрат на организацию информационного обеспечения, а неизбежного при использовании традиционных информационных технологий в новых экономических условиях.

Создание глобальной, централизованной, регулярно действующей сети циркулярной передачи данных решит ряд проблем укрепления внутренних связей в системе образования России и противостояния процессам дезинтеграции страны и регионов.

Российским Центром Информатизации Образования (РОСЦИО) совместно с инновационной компанией «Панорама» разработана концепция информационной системы компьютерных телекоммуникаций «ТВ-информ-образование». Научные руководители проекта д-р. техн. наук, профессор Я. А. Ваграменко и канд. техн. наук В. К. Сарьян.

Сеть «ТВ-информ-образование» предназначена для передачи информации учебным заведениям, органам управления, прочим учреждениям и организациям системы образования России. Основной задачей сети является повышение качества информационного обеспечения системы образования за счет увеличения объема передаваемой информации, сокращения времени доставки информации, уменьшения расходов на доставку. Сеть включает в себя приемные абонентские пункты, областные и региональные пункты подготовки и передачи информации, центральные пункты подготовки информации и глобальной передачи, структуры технического обслуживания и организации сети. Техническое обеспечение сети составляют персональные компьютеры (желательно IBM-совместимые), существующая телевизионная передающая и распределительная сеть. Пользователей сети можно разделить на пользователей-абонентов (потребителей информации) и поставщиков информации, заинтересованных в ее распространении.

Пользователь-абонент должен обладать компьютером, обыкновенной телевизионной антенной и специальной приставкой. Передача основного объема информации происходит в режиме циркулярного вещания. Местные пункты передачи и некоторая часть пользователей оснащаются аппаратурой обратной связи с центральными органами сети. Глобальная сеть предлагаемого масштаба позволит удовлетворить часть информационных потребностей в области образования и других стран, входивших ранее в СССР, будет способствовать сохранению единого информационного пространства и укреплению традиционных связей.

Внедрение сети «ТВ-информ-образование» будет происходить в основном на коммерческой основе и не предполагает какого-либо государственного административного регулирования. Перспективы развития сети и ее аппаратно-программное обеспечение позволяют рассчитывать на достаточно быстрое и эффективное ее включение в международные информационные сети.

Учреждения системы образования, занятые непосредственно в организации учебного процесса (учебные заведения, местные органы управления образованием), являются, по преимуществу, потребителями информации, возникающей и/или хранящейся вне системы образования, или в ее центральных органах. Обратный поток информации несравненно меньше и организуется значительно проще. Кроме того, в абсолютном большинстве случаев реакция периферийного объекта отделена от приема информации значительным промежутком времени. По этим причинам первый необходимый этап развития глобальной информационной сети должен обеспечить прежде всего передачу информации из центра на места в режиме регулярного сеансового вещания. Предлагаемые в нашей стране в настоящее время альтернативные проекты компьютерных коммуникаций сходного назначения технически не обеспечены или/и требуют громадных и совершенно нереальных затрат на техническое и организационное обеспечение. Сеть «ТВ-информ-образование» за счет максимального использования существующих технических средств и функционирующих организационных структур сети телевизионного вещания позволяет построить с минимальными затратами и в короткое время глобальную информационную сеть, удовлетворяющую потребности системы образования.

Построение и реализация сети «ТВ-информ-образование» разбивается на несколько этапов. Всем этапам предшествует «нулевой цикл», включающий в себя формирование принципов отбора информации, опыт-

ную эксплуатацию аппаратно-программных комплексов, размещение заказов на производство аппаратных средств, создание структур организационного, технического, информационного и методического обеспечения потребностей сети. Проблемы нулевого цикла будут решены в основном к концу 1992 г., а промышленная эксплуатация сети начнется в январе 1993 г. На первом этапе становления сети основной упор делается на регулярное обеспечение абонентов циркулярной информацией, передаваемой из центра. Одновременно будет развиваться промежуточная структура сети, обеспечивающая передачу информации местного значения. По мере развития сети и возрастания уровня потребностей пользователей произойдет переход к этапу массового внедрения систем обратной связи, обеспечивающей эффективное взаимодействие абонентов с центральными и местными узлами сети.

Пользователи сети

«ТВ-информ-образование»

Пользователями сети «ТВ-информ-образование» являются учреждения, организации, лица, заинтересованные в отправке информации, и абоненты, принимающие информацию. Сеть ориентирована на обслуживание потребностей системы образования и основными поставщиками информации могут быть центральные и региональные органы управления образованием, научные и методические учреждения, любые организации и лица, полагающие, что их информация представляет определенную ценность и актуальность для системы образования. Российский Центр Информатизации Образования (РОСЦИО) и созданные им организации, а также региональные центры передач принимают на себя обязанности по анализу потребностей пользователей-абонентов, формированию пакетов информации для сеансов передач, взаимодействию с пользователями (получателями и отправителями информации).

По территориальному или адресному признаку выделения абонентов следует различать следующие типы сеансов:

- 1) глобальные (ориентированные на прием всеми абонентами);
- 2) региональные (местные);
- 3) ориентированные на функциональную группу пользователей;
- 4) ориентированные на группу, сформировавшую специальный запрос;
- 5) персональные.

Глобальные и групповые передачи готовятся центральными органами сети на основе анализа потребностей всех или некоторой группы (выделенной не по территориальному признаку) учреждений системы образования Российской Федерации. Очевидно, что

потребности в информации у различных функциональных групп сильно отличаются, значительная часть объема должна быть адресована конкретной группе, и, по существу, параллельно необходимо готовить и осуществлять несколько сеансов передач для разных функциональных групп. Региональные передачи удовлетворяют местные потребности, и их целесообразно готовить и осуществлять с помощью местных передающих центров — подразделений глобальной сети. Подготовка передач для групп, сформировавших запрос, и персональных передач возможна только при наличии отлаженной системы обратной связи. Естественно, что формирование системы обратной связи будет несколько запаздывать по сравнению с формированием системы глобальной передачи. Помимо выше перечисленных, возможно также осуществление передач, ориентированных на прием зарубежными абонентами.

Некоторые пользователи-абоненты (например, региональные центры подготовки и проведения передач, научные учреждения, региональные центры), являющиеся одновременно и поставщиками информации, заинтересованы в интенсивном двухстороннем объеме информацией. Для этой цели они будут снабжаться дополнительным набором аппаратных и программных средств.

Информация, передаваемая по сети «ТВ-информ-образование»

Информация, подлежащая передаче по сети, классифицируется по содержанию, источникам и методам получения информации и по техническим особенностям представления информации.

С точки зрения содержания, потенциальным пользователям сети «ТВ-информ-образование» будет предоставляться следующая информация:

законодательные и нормативные документы системы образования;

практические рекомендации по организации управления и хозяйственной деятельности учреждений системы образования в новых и постоянно меняющихся условиях;

информация о потенциальных партнерах среди российских организаций, объединений и предприятий, возможные источники дополнительных ресурсов (финансовых средств, методической помощи, информационного обеспечения, программно-инструментальных и технических средств и т. д.);

учебно-методические материалы, педагогические инновации и новые информационные технологии, используемые, рекомендуемые или предназначенные для системы образования;

отечественный и зарубежный опыт приме-

нения учебно-методических материалов, педагогических инноваций и новых информационных технологий в системе образования;

обзоры и экспертные оценки состояния и перспектив развития системы образования, новых учебно-методических материалов, исследований в областях, связанных с системой образования;

государственные, региональные и прочие мероприятия системы образования (совещания, конференции, семинары);

государственные, региональные и прочие программы научно-педагогических, методических и других исследований, связанных с проблемами образования;

состояние рынка услуг системы образования, технические и программные средства обучения, учебно-методическая литература (в том числе — рекламные материалы);

технические и программные средства поддержки новых информационных технологий, требования и стандарты для аппаратных и программных средств.

По специфике и характеру источников поступления информация классифицируется следующим образом:

административная и нормативная (в том числе и учебно-методические и прочие материалы, рекомендованные к использованию органами управления образованием, информация о событиях в системе образования — конференции, семинары и т. д.);

научно-педагогическая, методическая и специальная информация, используемая в учебном процессе;

информация о новых информационных технологиях, применяемых в системе образования (в том числе и педагогические программные средства);

реклама и другая информация о рынке для системы образования;

прочая информация.

Особое внимание следует обратить на зарубежные источники информации, так как в настоящее время доступ к этой информации для широкого круга пользователей весьма затруднен.

Сеть «ТВ-информ-образование» будет представлять для абонента ценность и оправдывать вложенные им в аппаратно-программное обеспечение и абонентскую плату средства в том случае, если она в состоянии решить, по крайней мере, одну из следующих задач:

обеспечивать быстрое и регулярное удовлетворение потребностей в информации;

своевременно удовлетворять потребности в информации, трудно доступной по другим информационным каналам;

быстро ориентировать в окружающем мире, информационных потоках, рыночной ситуации;

обеспечивать более экономное (по сравнению с другими каналами) расходование средств на приобретение информации.

Значительная часть передаваемой информации будет состоять из документов и материалов, циркулирующих в центральных органах системы образования. Это разнообразные акты, учебно-методические материалы и компьютерные программы (в том числе и педагогические программные средства), распространяемые и рекомендованные министерством, ведомствами и учреждениями системы образования. Другие трудно доступные или дорогие учебные, методические, справочные и т. д. материалы, некоторые программные продукты будут приобретаться у их владельцев для распространения по сети из средств фонда, формируемого из абонентской платы. Возможность за сравнительно небольшую плату пользоваться этими источниками, а также информацией из банков даст абонентам сети прямую финансовую выгоду. Часть передаваемой информации составят обзорные, реферативные, библиографические материалы, подготавливаемые квалифицированными специалистами по заказу «ТВ-информ-образования». Важное место в потоке информации займут материалы, извещающие абонента о наличии на рынке, стоимости и условиях приобретения литературы, учебно-методических материалов, технических средств, программных продуктов, услуг, имеющих отношение и представляющих интерес для системы образования. Значительный резерв представляют научно-исследовательские программы, финансируемые из централизованных источников, результаты которых в настоящее время используются недостаточно эффективно.

Интенсивность передачи (частота и продолжительность сеансов) определяется в основном возможностями по подготовке информации, которые, в свою очередь, определяются объемом вложенных в подготовку средств и количеством сотрудников, привлекаемых для редакторской деятельности. Анализ потенциальных источников информации, технических возможностей и перспектив совершенствования передающей и принимающей аппаратуры позволяет сделать вывод, что после преодоления первых трудностей сеть выйдет в режим саморазвития, а увеличение числа абонентов и абонентской платы даст возможность вкладывать все больше и больше средств в развитие сети (прежде всего — в систему подготовки информации) и, как следствие этого, будет увеличиваться привлекательность сети для абонентов и их число.

Значительные проблемы на первоначальных этапах создает необходимость диффе-

ренцированного подхода к абонентам и подготовки одновременно нескольких сеансов передач, ориентированных на различные группы пользователей. По мере увеличения числа абонентов, развития структур, занятых поиском и подготовкой информации, будут увеличиваться и частота, и уровень специализации сеансов передач. Информация, передаваемая по сети, может представлять собой:

текстовые файлы (ASCII-коды), файлы с более или менее жесткой структурой (например, информационные и индексные файлы баз данных или архивированные файлы),

файлы видеоизображений, двоичные файлы (программы).

Передача файлов разного типа имеет некоторые особенности, но в конечном итоге абонент получает файлы в привычном и удобном для него виде. Прием информации осуществляется с помощью специализированного программного обеспечения, поставляемого вместе с аппаратными средствами.

Технические характеристики системы и аппаратуры

В настоящее время для системы передачи данных обычно используют разветвленные телефонные аналоговые сети, однако при этом сигналы передачи данных должны быть преобразованы в аналоговые сигналы с помощью модемов, что ограничивает скорость передачи данных. Для телефонных сетей общего пользования эта скорость не превышает 1200—2400 бит/сек.

В нашей стране использование телефонных линий для систем передачи данных ограничивается также катастрофическим дефицитом информационной емкости каналов и их низкой надежностью.

Предлагаемая система «ТВ-информ-образование» позволит прежде всего вывести из дуплексных систем передачи данных циркулярную информацию, что одновременно приведет к снижению затрат и улучшению условий эксплуатации сети. Для этой цели в системе рационально используется ТВ-вещательная распределительная сеть.

В настоящее время ТВ-вещанием охвачено 99,9 % населения страны (включая и бывшие союзные республики), наши распределительные ТВ-системы доставляют сигналы ТВ через спутниковые системы в любую точку страны и большинство стран земного шара.

Эти сети отличаются хорошим качеством, квалифицированным обслуживающим персоналом и существенно большей, чем в телефонии, надежностью.

Разрабатываемая система «ТВ-информ»

позволяет широко использовать возможности ТВ сетей по распределению дополнительной информации. Система рассчитана на распределение по адресным признакам текстовых и факсимильных сообщений, графических (включая полутоновые) и текстовых изображений.

По сравнению с другими системами уплотнение ТВ сигналов (например, система «Телетекст») «ТВ-информ» имеет ряд преимуществ: повышенную достоверность приема информации, меньшую зависимость качества приема цифровой информации от условий приема ТВ сигналов, возможность работы с различными конечными средствами документирования распределенной информации.

Система строится на базе единого ряда тактовых частот, один из уровней которого совпадает с тактирующей частотой «Телетекста». В основном варианте «ТВ-информ» применяется тактовая частота 1,73 МГц (1/4 тактовой частоты «Телетекста»). Сигнал строки данных состоит из последовательностей тактовой синхронизации по словам, адресам источника и непосредственно информации.

При выборе основного стандарта пропускная способность одной строки и кадр составляют 1230 Бод. При этом синхронизация по словам осуществляется с помощью кодирования по Баркеру, а в адресной последовательности используется кодирование по Хемингу. В системе «ТВ-информ» предусматривается возможность увеличения тактовой частоты в информационной части строк данных. Например, когда передаются избыточные сигналы (изображение, графика), то скорость может быть увеличена вдвое.

Структура системы циркулярной передачи данных может быть описана следующим образом. Информация от различных источников по соединительным линиям поступает в технический центр «ТВ-информ». Далее ТВ-сигнал, введенный в его состав дополнительной информацией через наземные и спутниковые линии связи, поступает на наземную телевизионную вещательную передающую сеть (ТВ-передатчики, ТВ-ретрансляторы) или в систему непосредственного спутникового ТВ-вещания, а затем через обычную ТВ-антенно-фидерную и кабельную распространительную систему на абонентские приемные устройства. К абонентским приемным устройствам могут подключаться ЭВМ различных типов, дисплеи, принтеры, факсы, телетайпы, аппараты воспроизведения изображений и прочие устройства регистрации информации.

На основе анализа адресной части строки данных и передаваемых в информационной

части адресов абонентов абонентские приемные устройства выводят потребителю предназначенную для него информацию.

Адресование происходит по нескольким уровням обобщения: единый, циркулярный, региональный, групповой, индивидуальный. Предусмотрена возможность распределения информации к потребителям в различных сочетаниях. Например, может быть реализована адресация на основе анализа квалификационных признаков информации, передаваемой вместо адресов.

При модификации строк данных, когда тактовая синхронизация и синхронизация по словам в каждом импульсе гашения полей передается однократно, можно повысить пропускную способность информационной части сигнала. Система «ТВ-информ» позволяет распределять информацию для различных пользователей, в том числе для системы «Телетекст». Информация «Телетекста» в основном формируется местными редакторами. В то же время есть ряд страниц общего содержания, передаваемых из центра. Одновременная оперативная доставка этих материалов всем местным редакциям может быть выполнена системой «ТВ-информ». По такой же схеме может формироваться центральная и местная информация в системе «ТВ-информ-образование».

Как правило, информация, передаваемая в «Телетексте», не имеет отношения к ТВ-программе. Потеря такой информации, например, при записи на видеомэгнитофон, при неудовлетворительных условиях эфирного приема не наносит ущерба при воспроизведении ТВ-программ. Дополнительная информация, введенная в систему «ТВ-информ», сохраняется в любой точке видеосигнала, т. е. помехоустойчивость сопутствующей информации должна соответствовать воспроизводимому ТВ-сигналу.

Введение сигнала дополнительной информации в модифицированный сигнал строк данных по частоте 1,73 МГц обеспечивает высокое качество приема дополнительной информации.

Удобство реализации заключается в сравнительно небольшом объеме сопутствующей информации, укладываемой в диапазоне 50—200 Бод (по сравнению с потенциальной пропускной способностью строки данных более 2400 Бод). Поэтому дополнительную информацию передают как часть общего потока и выделяют, используя признаки идентификации в определенных временных позициях сигнала строки (или модифицированного сигнала строк данных).

Разработанная базовая структура системы циркулярной передачи «ТВ-информ» может совершенствоваться. Аппаратура, ко-

торая начнет серийно выпускаться с ноября 1992 г., будет иметь недоступную на сегодняшний день для обычных массовых телефонных сетей достоверность доставки информации (вероятность ошибки 10^{-7} и выше), для чего будут использованы специальные помехоустойчивые способы передачи. В ряде случаев будет применяться дополнительное маскирование информации с использованием сменных ключей с целью предотвращения несанкционированного использования активной части растра (полностью или частично) для распределения информации во время перерывов передач. В том числе дополнительная информация может «вытеснять» часть служебного изображения, что позволит распределять информацию выше 1 Мбит/сек.

Особо отмечаем, что разработанная в системе «ТВ-информ-образование» система передачи данных обеспечивает одновременное распределение разнообразной информации от нескольких источников неограниченному количеству абонентов с доставкой данных по адресам. Это особенно важно, так как в системе народного образования сложилась ситуация, когда в школах, техникумах и вузах находится в эксплуатации ЭВМ и аппаратура различных модификаций.

Серийно выпускаемое приемное абонентское устройство выполнено комплектно на основе специально разработанного СВИСа и имеет габариты современного импортного видеоплеера. Это универсальное устройство может обеспечить прием информации на любое регистрирующее устройство, включая ЭВМ различных марок, со скоростью 4,8 Кбод, 9,6 Кбод, 1,54 Кбод. Для этого на передающей стороне будет предусмотрено соответствующее количество передающих устройств, которые будут передавать одну и ту же информацию для выделяемых по техническим характеристикам приемных устройств пользователей.

А/О «Панорама» и РОСЦИО предполагают обеспечить гарантийное и послегарантийное техническое обслуживание каждого абонента, независимо от того, где бы он не жил (в крупном городе или в маленьком поселке). ТВ-антенно-фидерная установка, абонентское приемное устройство и регистрирующее устройство в случае его неисправности будет заменено в течение 24 часов, а потерянная абонентом информация будет ему доставлена в те же сроки на удобном для

него носителе (дискете, бумажном носителе и т. д.).

Важно также подчеркнуть, что система «ТВ-информ-образование», являющаяся многоаспектной информационной системой циркулярной передачи данных, представляет возможность абонентам сети «ТВ-информ-образование» на приобретенных технических средствах стать абонентом и других информационных систем, которые развиваются параллельно с «ТВ-информ-образованием», а именно систем:

«ТВ-информ-банки» (банковская информация и компьютерная информация);

«ТВ-информ-бизнес» (биржевая, нормативная и компьютерная информация);

«ТВ-информ-патент» (патентная информация, полное описание и графическая иллюстрация ретроспективных и новых патентов, созданных в стране и за рубежом, нормативная патентная документация, регулирующая авторские права изобретателя);

«ТВ-информ-экология» (нормативная информация по экологическим вопросам, экспертные экологические системы, информация об экологическом образовании, инструкции по привлечению абонента к участию в экологическом эксперименте или мониторинге);

«ТВ-информ-нормативы» (нормативная и текущая информация таможенного комитета РФ и налоговой инспекции РФ и другая циркулярная информация, полезная для абонента).

При этом абонент только вносит соответствующую дополнительную абонентскую плату за участие в той или иной информационной сети.

Использование в системе «ТВ-информ-образование» унифицированной приемной абонентской аппаратуры позволяет существенно снизить эксплуатационные расходы на ее содержание и переложить часть затрачиваемых средств на другие, приемлемые по иным соображениям, информационные сети, что имеет немаловажное значение для системы народного образования.

Все работы, связанные с формированием, функционированием и развитием сети, координируются РОСЦИО, который обеспечивает соответствие передаваемой информации требованиям, нормам и потребностям системы образования. Телефоны для справок: (095) 928-87-14, 924-98-93.

Основные задачи: организация и координация работ в области информатизации учебного процесса, науки и управления системой образования, создание банка педагогических данных.

Функции РОСЦИО: разработка научных основ единой политики в области компьютеризации и внедрения информационных технологий в системе Министерства образования РФ;

организация и проведение работ по государственным программам информатизации образования, формирование сети региональных центров информатизации образования; научно-методическое руководство и координация научных исследований по проблемам информатизации, проводимых учебными заведениями системы Министерства образования РФ;

развитие и поддержка отраслевых баз данных, разработка их структуры, содержания и системное построение;

экспертиза разрабатываемых программно-технических средств, пропаганда передовых информационных технологий в образовании; формирование Российского фонда компьютерных учебных программ;

разработка программно-информационного инструментария для изучения и моделирования с помощью ЭВМ структур образования и управления образованием;

повышение квалификации и подготовка кадров в области информатизации отрасли образования.

Информационные структуры и системы РОСЦИО

Система «ТВ-информ-образование». созданная при научно-техническом сотрудничестве с компанией «Панорама», — высокоскоростной и надежный канал распространения цифровой информации путем уплотнения сигналов в существующих ТВ-вещательных сетях; передача и прием на компьютер из эфира текстов, пакетов прикладных программ, графической информации; два уровня маскирования информации с применением индивидуальных ключей на приемной стороне у абонентов (несанкционированный прием исключается), организация ре-

гиональных пунктов, участвующих в формировании общей программы компьютерных телекоммуникаций посредством «ТВ-информ-образование» на всей территории России. Возможности, достигнутые в «ТВ-информ-образовании», пока не способны обеспечить какие-либо другие технические решения.

Российский фонд учебных компьютерных программ обеспечивает координацию и научно-методическую поддержку сети региональных фондов, единую нормативную базу фонда, создание и функционирование интегрированной информационной системы распределения фонда и ведение центрального фонда компьютерных образовательных программ, государственную сертификацию и экспертизу учебного программного продукта, мобильное представление программ потребителям.

Банк педагогических данных имеет в своем активе базы данных по проблемам: дошкольная педагогика, общеобразовательная школа, социальная педагогика, непрерывное образование, подготовка кадров, педагогические изобретения и открытия, управление педагогическими системами и др. В каждой базе данных упорядочена следующая информация: концепция, теория, модели, алгоритмы, прикладные данные, разработки, передовой и отрицательный опыт, нормативы, законы, рекомендации, нерешенные проблемы, запросы на педагогическую информацию. Сведения основаны на отечественной и зарубежной науке и практике.

Банк данных социально-психологического и медицинского обследования населения «Дети Чернобыля» — впервые созданная интегрированная многопрофильная информационная система для отображения состояния и анализа развития детей, проживающих в зонах экологического неблагополучия, и в особенности при наличии последствий радиационного заражения. Этот банк данных имеет особое значение для учреждений образования на соответствующих территориях, выработки и осуществления социальных программ.

Местонахождение РОСЦИО: 103031, Москва, ул. Рождественка, 6/20, тел. дирекции: 928-87-14, факс: 925-29-69.

Официальный дилер ПО "Сетман" -
фирма "Коллеж" - реализует в Москве

КОЛЛЕЖ

УЧЕБНЫЕ КОМПЬЮТЕРЫ И ОРГТЕХНИКУ

Учебный класс "Искра-1031"

С марта 1991 года установлено более 500 классов в России и странах содружества. Развернуто более 150 центров по обслуживанию техники. Вместе с техникой поставляются пакеты учебных программ. Учебный класс на базе "Искра-1031" включает рабочее место преподавателя и до 12 рабочих мест ученика, объединенных локальной сетью. Библиотека учебных программ хранится на рабочем месте преподавателя. "Искра-1031" программно совместима с IBM PC/XT.

ПОРТАТИВНАЯ ГИШИЦА МАШИНА «ИВИЦА»

В машине используются стандартные чашечные литероносители.
• Длина печатаемой строки, мм 215
• Скорость печати, зн/сек до 10
• Ширина рабочей ленты, мм 13

ПЭЛТ-305-02 «РОМАШКА»

Скорость печати, зн/сек - 12.
• Средне-узкая память на 4 текста (2 стр. печатного текста)
В машине используются сменные лепестково-дисковые литероносители. Возможен выпуск с алфавитом языков СНГ.

Персональный компьютер "Искра-3104"

"Искра-3104" - первый российский аналог IBM PC/XT TURBO, совместимый не только программно, но и конструктивно на уровне разъемов и слотов с зарубежным аналогом. Благодаря этой совместимости "Искра-3104" может оснащаться любыми импортными принтерами, модемами и другими периферийными устройствами. Вместе с компьютером поставляются пакеты программ делового и административного назначения. Можно использовать любые программы для IBM PC.

ЭЛЕКТРОННЫЙ СЕКРЕТАРЬ

Это малогабаритная приставка к Вашему компьютеру, которая в нужное время напоминает Вам о важных делах, позвонит в указанное время по телефону, передаст устные сообщения, поможет Вам записать телефонные разговоры, а затем воспроизведет их.

По вопросам поставки обращайтесь по адресу: Москва, Волковский пер. д.11, фирма "Коллеж". Тел. 267-70-58

В. ТИЩЕНКО, Г. ТИЩЕНКО

Киев

Что делать, если отказал компьютер?

Современный персональный компьютер состоит из миллионов мельчайших электронных элементов (в основном транзисторов). Достаточно одному транзистору выйти из строя — и может наступить сбой в работе всего ПК. Понятно, что отыскать неисправный транзистор среди миллионов ему подобных, а тем более заменить его на исправный — дело очень и очень трудное. И хотя реально меняют не транзистор, а целую микросхему, трудности ремонта усугубляются тем, что ПК, в отличие от других электронных изделий (телевизор, магнитофон и т. п.), работает по программе, размещенной в его памяти. Такой режим работы вызывает особый вид «программных» неисправностей. Программные и аппаратные (технические) неисправности бывают настолько взаимосвязаны, что, оказывается, очень трудно их различить и установить причину отказа.

Вскрыв компьютер, вы увидите десятки, а иногда и сотни деталей: интегральных микросхем, резисторов, конденсаторов и т. д. Любая из них может выйти из строя, и для начала мы расскажем о характерных причинах этого.

Интегральные микросхемы (плоские пластмассовые корпуса с большим числом металлических ножек-выводов) содержат от десятков до десятков тысяч транзисторов, внешне различаются числом выводов. МИС (малые интегральные схемы, десятки транзисторов) имеют 14—16 выводов, СИС (средние, сотни и тысячи) — 20—24, а БИС (большие, десятки и сотни тысяч) — 40 и более. Часто БИС устанавливают на плату с помощью переходных разъемных панелек. Такой способ облегчает замену неисправных БИС; в то же время плохое качество переходных контактов часто является причиной неисправностей.

Размеры транзисторов в ИС чрезвычайно малы, и, хотя проходящий через них ток тоже невелик, они заметно нагреваются; в период же отсутствия сигнала они начинают остывать. Такой многократно повторяющийся процесс нагревания-остывания получил название термического стресса и часто служит причиной выхода ИС из строя. Чтобы уменьшить влияние термического стресса, следует располагать ПК достаточно далеко от нагревательных приборов, а также не поль-

зоваться им слишком долго без отдыха*. Нужно также обеспечить достаточный доступ воздуха для охлаждения ИС, расположенных на плате.

Другая причина отказов ИС — явление, называемое миграцией металла. Плотность тока (ток, деленный на площадь сечения) в сверхминиатюрных проводниках настолько велика, что молекулы металла «сдувает» «электронным ветром». Следствием миграции могут стать разрыв проводников, загрязнение примесями полупроводниковых элементов и т. п.

Следующая проблема возникает при включении и выключении питания. Некоторые ИС (например, микропроцессор КР580ИК80А, микросхема памяти К565РУ3 и др.) требуют соблюдения определенной последовательности при подаче питающих напряжений — иначе возможен их выход из строя. Первым должно включаться напряжение $-5В$, затем $+5В$ и, наконец, $+12В$; выключение производится в обратном порядке. Допускаются одновременные подача или выключение всех напряжений. Если почему-либо пропало одно из напряжений, немедленно должны быть выключены и остальные. Особенно опасно отсутствие напряжения $-5 В$.

Известную угрозу создают также отклонения величины питающего напряжения. Допускаются колебания в пределах $\pm 5\%$. При меньшем напряжении ИС будет давать сбой, при большем может выйти из строя.

Часто причиной неисправной ИС является замыкание выводов.

Многие ИС (например, типа к-МОП) боятся электростатических зарядов. Эти заряды легко возникают при трении современных синтетических материалов и могут достигать сотен и тысяч вольт, в то время как рабочее напряжение ИС всего несколько вольт. Чтобы исключить попадание электростатических зарядов на выводы ИС, нужно при вскрытии компьютера заземлять рабочий инструмент и себя (одевая на руки заземленный браслет и т. п.).

Часто причиной неверной работы ИС является отсутствие контакта между ее выво-

* Распространено и другое мнение: компьютер следует выключать как можно реже (только на ночь). Оно основывается на том, что блок питания выдает в момент включения опасный электрический импульс. — *Примеч. ред.*

дом и проводником на печатной плате (плохая пайка) или контактным разъемом переходной панели.

Резисторы различных типов могут иметь следующие неисправности:

изменение сопротивления вследствие подгорания рабочего слоя;

механические повреждения (трещины, царапины, излом);

бесконечно большое сопротивление вследствие обгорания рабочего слоя или отсутствия контакта в месте пайки выводов.

У переменных резисторов зависимость сопротивления от положения движка может быть неплавной.

Конденсаторы (бумажные, слюдяные, керамические и др.) постоянной емкости могут иметь следующие неисправности:

пробой (короткое замыкание между пластинами);

обрыв выводов, плохой контакт в месте пайки;

утечка заряда.

Характерными неисправностями диодов являются:

пробой (ток в прямом и обратном направлениях проходит без сопротивления);

обрыв вывода, плохой контакт в местах пайки;

изменение электрических параметров, часто вследствие перегрева при пайке или эксплуатации (увеличение обратного тока).

Биполярный транзистор можно упрощенно представить в виде двух диодов, включенных, как показано на рис. 1. Поэтому для транзисторов характерны те же неисправности, что и для диодов:



изменение электрических параметров;

пробой эмиттерного, коллекторного или обоих переходов, проявляющийся в виде короткого замыкания или бесконечно большого сопротивления;

плохой контакт в местах пайки выводов. Многоконтактным разъемам присущи следующие неисправности:

механическое повреждение штыря или гнезда;

неправильное включение ответной части (включен лишь один ряд вместо нескольких, разъем смещен на несколько штырей в сторону, поворот ответной части на 180° и др.);

короткое замыкание соседних контактов разъема;

неполное включение (не до упора) штыревой части в гнездовую;

обрыв проводников кабеля разъема.

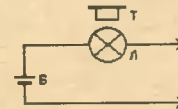
Для проведения профилактических и мелких ремонтных работ необходимо иметь набор отверток (в том числе крестообразных), пинцеты, кусачки (бокорезы), плоскогубцы, паяльник. Особое внимание следует уделить паяльнику. Учитывая, что плата и расположенные на ней детали (особенно ИС) боятся перегрева, а выводы деталей расположены очень тесно, необходим миниатюрный мало-мощный паяльник с тонким жалом. В то же время для выпаивания ИС из платы пригодится более мощный паяльник с относительно толстым жалом и специальной насадкой, позволяющей одновременно выпаять целый ряд выводов.

Для снятия статических зарядов необходимо все инструменты соединять гибким проводом через резистор, имеющий сопротивление 1 МОм, с заземлением.

Тестером обычно называют «тройной» измерительный прибор: амперметр — вольтметр — омметр, позволяющий измерять силу тока, напряжение и сопротивление электрической цепи. Для поиска неисправностей он необходим.

Современные тестеры (имеющие цифровые индикаторы, а не стрелочные приборы) обладают очень большим входным сопротивлением, благодаря чему подключение их к электрической цепи не влияет на ее характеристики, чем обеспечивается высокая точность измерений.

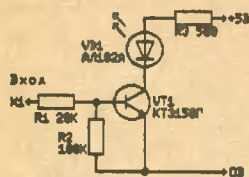
Простейший пробник. При испытании электрической цепи и деталей схемы на обрыв или короткое замыкание часто вместо тестера пользуются простейшим пробником, состоящим из последовательно соединенных лампочки на 2,5 В и одного элемента 373 (рис. 2); вместо лампочки можно включить



головные телефоны. С телефонами вместо батарейки в качестве источника напряжения можно использовать выходное звуковое напряжение радиоприемника, магнитофона или розетки радиотрансляционной сети.

Логический пробник позволяет определять уровень логического напряжения в нужной точке электронной схемы. Уровень может быть низким (0) или высоким (1), а может и отсутствовать (обрыв цепи). Простейший пробник индицирует логический уровень при помощи одного светодиода: отсутствие свечения — низкий уровень, яркое свечение — высокий уровень, тусклое свечение — отсутствие логического уровня.

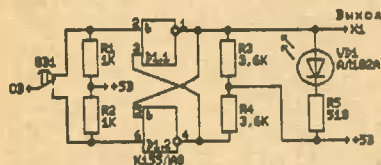
Конструктивно пробник выполняется в виде стержня, напоминающего шариковую авторучку, на конце которого расположен измерительный щуп, а рядом — светодиод. Его схема приведена на рис. 3.



Когда на вход X1 подается уровень логического нуля, транзистор VT1 закрывается, ток через него прекращается и светодиод VD1 гаснет. При подключении к X1 уровня логической единицы транзистор открывается и протекающий ток вызывает яркое свечение светодиода.

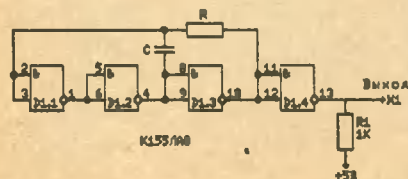
38 Источник (генератор) логических сигналов. Источником логического сигнала низкого уровня (0) может служить провод, соединенный с общей точкой (ОВ) питающих напряжений, а источником высокого уровня (1) — провод, соединенный с +5 В через ограничивающий резистор (для ИС серии 155R=1 КОм).

Недостатком этих источников является дребезг контактов при подключении и отключении проводников. Для устранения дребезга часто используют схему RS-триггера, приведенную на рис. 4. В исходном состоянии



кнопка SB1 отжата и на выход X1 подается уровень логической единицы. При нажатии SB1 триггер, собранный на элементах DD1.1 и DD1.2, опрокидывается и на X1 появляется уровень логического нуля. Изменение уровней на X1 происходит мгновенно, без дребезга. Диод VD1 служит индикатором выдаваемого уровня.

Если требуется источник, способный выдать серию сигналов, то применяют схему, изображенную на рис. 5. Частота и длительность выдаваемых сигналов зависит от величин C и R.



Осциллограф является очень удобным универсальным прибором. Он позволяет измерять многие параметры электрических сигналов и наглядно наблюдать их изменение во времени.

Недостатками осциллографа являются относительная дороговизна и необходимость довольно длительного обучения работе с ним.

Логический анализатор — это многоканальный (8, 16 и даже 64 канала) осциллограф). Он с большой скоростью поочередно подключается к исследуемым каналам, а результаты измерений запоминает в ячейках памяти. Затем данные из памяти выводятся на экран, отображая согласованные картины состояний всех исследуемых каналов.

Сигнатурный анализатор дополнительно снабжается генератором так называемого сигнатурного кода — программируемой последовательности нулей и единиц, подаваемой на вход исследуемой электронной схемы, в то время как многоканальный осциллограф анализатора подключается к ее ключевым точкам. По данным кодов на входе и выходах судят о состоянии схемы.

Логический и сигнатурный анализаторы — сложные и дорогостоящие приборы, используемые в условиях специализированных мастерских.

Алгоритм устранения отказа

Перейдем к рассмотрению главного вопроса — как и в какой последовательности действовать, чтобы установить место неисправности и устранить ее. Шагами соответствующего алгоритма являются:

- выявление признаков неисправности;
- определение ее места и причины;
- устранение.

Выявление признаков неисправности. Признак неисправности — это те (не всегда ясно видимые) проявления, в работе являются следствием нарушений в работе ПК. Подобно тому как больной человек может испытывать лишь легкое недомогание или находиться в состоянии тяжелого шока, ПК может по-прежнему выполнять свои функции, хотя и не так хорошо, как прежде, а может и полностью отказать.

При логическом анализе нужно иметь в виду следующие возможные виды неисправностей:

• полный и постоянный отказ в работе (простейший вид);

• наблюдаемый дефект является нежелательным, но позволяет продолжать работу (пониженная яркость экрана дисплея, тугой ход одной из клавиш и т. д.);

• неисправность затрудняет, а в отдельных случаях делает невозможным выполнение работы;

• дефект появляется периодически: после пе-

риода нормальной работы возникает отказ, а спустя некоторое время работоспособность восстанавливается (часто это бывает при нагревании деталей и плат ПК; после остывания неисправность исчезает).

Техническая диагностика имеет много общего с медицинской. Подобно тому как при заболевании человека необходимо знать симптомы болезни, так и при поиске неисправностей в работе ПК важно точно знать их признаки.

Определение места и причины неисправности — одна из труднейших задач. Этот процесс требует порой сложнейшей аппаратуры, создания и аналитического исследования специальной математической модели отказа. Такие способы поиска можно применять только в специализированных мастерских. В данной статье мы укажем лишь на некоторые простейшие приемы поиска неисправности.

Прежде всего отметим многоальтернативный, ветвящийся характер поиска. Используя дерево логических возможностей (рис. 6), удастся постепенно сократить число предположений о месте отказа и быстрее обнаружить отказавший элемент.

мально, а одну со сбоем, то скорее всего причина — ошибка, имеющаяся в этой программе, или использование «чуждой» версии языка программирования, неподходящей версии ОС и т. п. Возможны и повреждения носителя программы (дискеты, магнитофонной кассеты, микросхемы ПЗУ), а также влияние компьютерного вируса (КВ).

В связи с началом широкого распространения КВ расскажем о них подробнее.

КВ — злоумышленно разработанная и тайно внедренная специальная программа, способная к самораспространению без ведома и вопреки желанию пользователя. Под воздействием КВ может входить в бесконечный цикл перезагрузки, демонстрировать «осыпание» букв на экране, стирать во всех файлах какое-либо число (например, 13), издавать непонятные звуки, уничтожать всю информацию, накопленную в ОЗУ, производить целый ряд других действий.

КВ обычно попадает в компьютер при использовании случайных программ, одалживаемых у знакомых для перезаписи, или же при использовании своих дискет на зараженном КВ компьютере.

Самым простым способом обнаружения КВ является анализ объема памяти, занимаемого рабочей программой. Если известно, сколько килобайт занимала программа ранее и сколько она занимает теперь, то изменение объема (чаще всего увеличение) будет свидетельствовать о заражении программы вирусом.

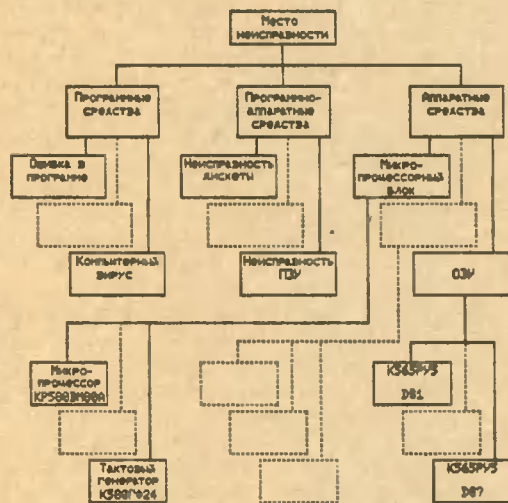
Вернемся к анализу правой, аппаратной части дерева.

Если экран дисплея не светится, вентилятор охлаждения не вращается, а светодиоды, сигнализирующие о включении питания, не горят, то логично предположить, что программное обеспечение здесь ни при чем. Очевидно, перегорел предохранитель в цепи питания или просто вилка сетевого шнура плохо вставлена в розетку.

Учитывая, что аппаратная часть ПК является системой функционально связанных блоков, разделяют систему на блоки и, зная признак неисправности, сопоставляют его с функциональным назначением блоков. Таким образом удастся приблизительно выделить неисправный блок. Состояние блока оценивают путем внешнего осмотра его деталей, поиска обрывов и коротких замыканий в его монтаже, а также путем измерения характеристик их и сравнений с номинальными значениями.

Если заменить подозреваемый блок заведомо исправным, то можно окончательно убедиться в неисправном состоянии заменяемого блока.

Большую помощь в определении места и причины неисправности могут оказать встроенные в ПК (в его ПЗУ) или записанные



На рис. 6 приведена лишь общая схема построения такого дерева. В реальной обстановке оно может быть намного сложнее.

Поиск начинают с логических рассуждений над верхней (программно-аппаратной) частью дерева, затем опускаются ниже и пытаются выявить возможность повреждения функционального блока или узла. Далее, зная устройство подозреваемого блока и принцип его действия, стараются определить, какой из элементов блока может быть причиной отказа.

Если ПК обрабатывает все программы нор-

на дискетах специальные диагностические программы, выдающие на экран дисплея условный код неисправности или подробный текст, описывающий неисправность и даже указывающий конкретно неисправный элемент. Понятно, что польза от диагностической программы будет лишь в том случае, если ПК хотя бы отчасти работает.

Помимо диагностических программ иногда используют пошаговый режим работы ПК, позволяющий проанализировать действия блоков и узлов компьютера в замедленном темпе. Пошаговый режим осуществляется при помощи программных или аппаратных средств (специальной электронной приставки).

Некоторые типы компьютеров (БК-0010, «Спектрум» и др.) конструктивно выполнены так, что очень трудно выделить функциональный блок. На одной печатной плате этих компьютеров смонтированы микропроцессор, ОЗУ, ПЗУ, контроллеры дисплея и магнитофона и др. Сложность и неразделимость блоков в таких компьютерах осложняют поиск неисправности.

На практике почти не встречается выход из строя всех или очень большого количества элементов ПК (исключая случаи возгорания или падения ПК с высоты). Чаще всего отказывают один или несколько связанных друг с другом элементов. Поэтому поиск отказа в большинстве случаев сводится к поиску конкретного неисправного элемента ПК.

Проверка элементов ПК. При проверке элементов, впаянных в плату, нужно учитывать шунтирующее влияние параллельно подключенных электрических цепей и других элементов схемы. Если такое влияние велико, то перед измерением следует отсоединить выводы проверяемого элемента от общей схемы. Это обычно сопряжено с трудностями. С помощью ножа или скальпеля можно перерезать печатный проводник, но восстановить соединение порой не удастся.

При подаче измерительных сигналов нулевого и единичного уровня в контрольную точку схемы необходимо учитывать состояние электронных элементов, подключенных к ней. Если логическая ИС выдает единичный уровень, а пробником в эту же точку схемы подан нулевой уровень, выходной транзистор ИС может выйти из строя. По этой же причине нельзя при испытаниях соединять параллельно выходы ИС. Если одна ИС будет выдавать нуль, а вторая — единицу, то может возникнуть недопустимо большой ток и ИС выйдет из строя. Параллельно можно соединить лишь выходы ИС, имеющие так называемый открытый коллекторный выход (например, K155ЛА8).

Перед электрической проверкой (измере-

нием) необходимо осмотреть монтаж и элементы на печатной плате. Монтаж можно осматривать при включенном или при выключенном питании ПК. В первом случае задача осмотра состоит в определении неисправного места по таким признакам, как пробой напряжения на землю, чрезмерное нагревание элементов, запах; во втором — в проверке целостности печатных проводников, отсутствия замыканий между выводами, надежности механического крепления элементов, их цвета и формы, качества пайки. Особое внимание следует обращать на пайки, отличающиеся неровной поверхностью, пористостью и малым количеством припоя.

Часто причиной неисправной работы элементов ПК является блок питания, поэтому проверку обычно начинают с измерения величины питающих напряжений. Измерение выполняют вольтметром непосредственно на выводах питания ИС, расположенных в различных местах платы.

Проверка ИС. Проверку логических микросхем типа И-ИЛИ-НЕ, триггеров, счетчиков, дешифраторов и некоторых других МИС производят с помощью таблиц истинности, в которых указывается, какие логические уровни должны быть на выходах МИС при тех или иных логических уровнях на входах (примеры приведены на иллюстрации). Уровни на входе задают с помощью источника логических напряжений, а на выходе наблюдают с помощью логического пробника, тестера или осциллографа.

При проверке логических МИС нужно иметь в виду, что, если на какой-либо вход логического элемента сигнал не подан, микросхема обычно «будет считать», что на нем установлена логическая 1.

Некоторые типы ИС работают при обрыве в цепи питающего напряжения, хотя и не являются «вечными двигателями». В этом случае работа происходит (часто со сбоями) за счет энергии входных сигналов.

Проверка БИС весьма затруднена из-за их сложности. Если БИС установлена на плате при помощи переходной панельки, то проверку осуществляют путем замены БИС на заведомо исправную, а если панелька отсутствует, то выполняют лишь частичную проверку путем измерения логических уровней на наиболее важных выводах.

Неисправность ИС определяют и по степени нагрева. Нормально работающая ИС обычно бывает слегка теплой. Слишком горячая или совершенно холодная ИС может оказаться неисправной.

Довольно часто сбой происходит из-за плохого контакта выводов ИС с гнездами переходной панельки. Необходимо тщательно проверить, правильно ли вставлена ИС в панельку (не повернута ли на 180°), не отогнут

ли один из выводов, нет ли перекосов, замыканий и механических повреждений выводов, плотно ли, до конца вставлена ИС в панельку.

Проверка постоянных резисторов различного типа производится внешним осмотром, приложением механических усилий к выводам и измерением с помощью омметра номинальной величины сопротивления. На поверхности резистора не должно быть трещин, царапин, подгорания и других дефектов.

Качество переменных проволочных и непроволочных резисторов определяется внешним осмотром, проверкой надежности контактирования и измерением величины сопротивления. При осмотре обращают внимание на состояние всех его наружных частей и плавность хода. Особенно тщательно проверяют плавность изменения сопротивления при вращении оси.

Проверка конденсаторов. Бумажные, слюдяные, керамические и другие неэлектролитические конденсаторы постоянной емкости подвергаются внешнему осмотру и проверке на отсутствие замыкания между обкладками, обрыв и утечку. Измерение емкости обычно не производят, так как для этого нужен специальный прибор.

При осмотре обращают внимание на отсутствие вмятин, изломов и прочность крепления в плате.

Проверку на короткое замыкание, обрыв или утечку лучше всего выполнять при помощи простейшего пробника (см. рис. 2). Если конденсатор исправен, то присоединение к нему щупов пробника вызывает щелчок в телефоне при первом замыкании цепи. Повторное касание не создает щелчка, так как конденсатор уже заряжен. Если конденсатор пробит или имеет обрыв, то в первом случае щелчки воспринимаются при каждом присоединении щупов, а во втором — ничего не слышно. Если конденсатор имеет утечку, то щелчки прослушиваются спустя несколько секунд; чем больше пауза, тем громче щелчок. При подозрении на изменение емкости конденсатор заменяют другим, заведомо исправным, имеющим нужную емкость.

Проверка электролитических конденсаторов выполняется при помощи омметра. При присоединении прибора стрелка быстро отклоняется, а затем медленно возвращается в исходное положение. Чем больше емкость, тем на больший угол отклоняется стрелка.

Если конденсатор потерял емкость или имеет значительную утечку, то стрелка почти не отклоняется или очень медленно и не полностью возвращается в исходное положение.

Электролитический конденсатор подвергается также внешнему осмотру, проверке крепления и отсутствия подтекания электролита.

Проверка диодов. Убедиться в исправности диода можно путем измерения его прямого и обратного сопротивлений.

Если прямое бесконечно велико или обратное равно нулю, диод неисправен.

Проверка транзисторов. Наиболее простой и широко распространенный способ проверки транзисторов — измерение прямого и обратного сопротивлений его эмиттерного (участок база—эмиттер) и коллекторного (база—коллектор) переходов. Обратное сопротивление должно быть во много раз больше, чем прямое. Если эти сопротивления равны нулю или бесконечности, транзистор неисправен. Производится также измерение сопротивления коллектор—эмиттер. Оно должно быть достаточно велико. Если оно близко к нулю, то транзисторы, установленные на радиаторах, проверяются на степень допустимого нагрева.

Устранение неисправности

Вышедший из строя элемент ПК обычно ремонту не подлежит. Его приходится заменять новым, заведомо исправным.

При замене элемента, расположенного на печатной плате, возникает значительные трудности с его демонтажом. Выпаять элемент, не повредив печатный монтаж, очень трудно, особенно при замене ИС, имеющих много выводов.

Если нет достаточного опыта такой замены, то не следует начинать демонтаж. Нужно предварительно приобрести опыт, заменив несколько ИС на старой, ненужной плате какого-либо электронного устройства.

Демонтаж начинают с отсасывания припой вокруг выводов ИС (например, с помощью «косички» из медных проводков, которые, будучи нагретыми, «впитывают» припой). Затем на жало паяльника одевают насадку, позволяющую одновременно прогреть весь ряд выводов ИС. При достаточном разогреве этот ряд освобождается довольно легко, после чего такую же операцию повторяют и над вторым рядом выводов.

Возможен и другой способ: откусывают выводы неисправной ИС у самого основания корпуса, а затем выпаивают их поочередно. После этого отверстия в плате очищают с помощью толстой иглы и вставляют в них выводы новой ИС.

Особенно трудно заменить элемент на плате, покрытой (в целях защиты от внешних воздействий) особым стекловидным лаком. Выпаять элемент из такой платы практически невозможно. Приходится откусывать выводы у основания негодного элемента, соскабливать с них лак, заслуживать и припаивать к ним выводы нового элемента.

Несколько правил для начинающих

Не пытайтесь обнаружить неисправность с помощью отвертки или пинцета. Непродуманные действия могут привести к печальным последствиям. Не зная, например, о влиянии электростатических зарядов на микросхемы, можно, орудия отверткой, случайно коснуться вывода микросхемы, внести этим статический заряд электричества и вывести микросхему из строя. Или другая опасность. В блоке дисплея, в цепи питания электронно-лучевой трубки, имеется конденсатор, сохраняющий заряд высокого напряжения в течение многих часов после выключения дисплея. Не зная этого, можно случайно коснуться цепи высоковольтного конденсатора и получить сильный удар электрическим током.

Электронные элементы ПК (микросхемы, резисторы, конденсаторы и др.) обладают высокой надежностью и механической прочностью. При нормальной эксплуатации они очень редко выходят из строя (особенно импортные элементы). Как показывает опыт, сбой в работе ПК чаще всего происходит по следующим причинам:

ошибочные действия персонала;
плохой контакт (обрыв) в электрической цепи;

короткое замыкание проводников;
механические повреждения;
неисправности в цепях питания;
неполадки программного обеспечения.

Поэтому, если что-то случилось с компьютером, не следует спешить с его вскрытием.

Необходимо внимательно подумать: не допустили ли вы какой-либо ошибки в своих действиях, все ли вы делаете так, как требует инструкция по эксплуатации? Избегайте поспешных действий!

Прежде всего убедитесь, что все блоки, входящие в состав ПК, надежно соединены друг с другом и все органы управления (тумблеры, кнопки, переменные резисторы и др.) находятся в положении, предусмотренном инструкцией по эксплуатации. Учитывая взаимозаменяемость многих разъемов, не лишне проверить, в свои ли разъемы включены кабели клавиатуры, дисковод и дисплея.

Полезно воспользоваться таблицей типовых неисправностей, обычно проводимых в инструкции по эксплуатации. Правда, целиком полагаться на эти таблицы не стоит, так как они часто пишутся формально и содержат много неточностей, двусмысленностей и оговорок. Если есть возможность, замените подозрительный блок заведомо исправным. Если в классе установлено несколько ПК и на одном из дисплеев пропало изображение, попробуйте заменить этот дисплей другим. Точно так же можно менять клавиатуру, дисковод и другие блоки ПК.

Главное, о чем следует помнить: задача состоит не в том, чтобы обязательно отыскать причину и устранить неисправность, а в том, чтобы не усугубить неисправность и не отправить в ремонт компьютер, у которого всего-то перегорел сетевой предохранитель или отпаялся провод от кнопки сброса.

М. КОВТУН

Программирование в системе Express Pascal

Условные операторы. Составной оператор

Вернемся к программе решения квадратного уравнения. Один из ее недостатков — то, что она не проверяет, имеет ли решаемое уравнение вещественные корни (если вещественных корней нет, возникает ошибка времени исполнения).

Программа решения квадратного уравнения с проверкой существования вещественных корней будет иметь следующий вид.

```
var a, b, c : real; { коэффициенты уравнения }
    D, SD : real; { дискриминант и
                  { квадратный корень из него }
```

```
x1, x2 : real; { корни уравнения }
begin
  Write('Введите коэффициенты
        уравнения a, b и c ');
  Readln ( a, b, c );
  D := Sqr(b) - (4*a*c);
  if D >= 0
  then begin
    SD := Sqrt(D);
    x1 := (-b + SD) / ( 2*a );
    x2 := (-b - SD) / ( 2*a );
    Writeln;
    Writeln('Корни квадратного уравнения:');
    Writeln(' x1 = ', x1 );
    Writeln(' x2 = ', x2 );
  end
  else Writeln('Уравнение не имеет
               вещественных корней');
end.
```

Продолжение. Начало см.: Информатика и образование. 1992. № 1—4.

Чтобы выполнить проверку, нам понадо-

бился условный оператор *if*. В общем случае оператор *if* имеет формат

```
if <логическое-выражение>
then <оператор1>
else <оператор2> ;
```

Логическое выражение строится способом, очень похожим на способ построения арифметического выражения: из констант и переменных логического типа и элементарных логических выражений с помощью знаков операций *not*, *and*, *or* и *xor*. Любое логическое выражение принимает одно из двух значений: «истина» или «ложь».

Имеются две константы логического типа: *true* и *false*, значениями которых являются «истина» и «ложь» соответственно. (С точки зрения синтаксиса *true* и *false* являются предопределенными идентификаторами.)

Переменная логического типа — это переменная, тип которой есть *boolean* (это тоже предопределенный идентификатор). Например, следующее описание определяет переменную *flag* как логическую:

```
var flag : boolean;
```

В программах логические константы и переменные используются относительно редко; в основном используются логические выражения, построенные из элементарных логических выражений. Элементарное логическое выражение строится из двух логических выражений с помощью операций сравнения:

```
= «равно»
> «больше»
< «меньше»
>= «больше или равно»
<= «меньше или равно»
<> «не равно»
```

(Обратите внимание на то, что знаки последних трех операций составлены из двух литер. В их записи нельзя ставить пробел между литерами. Например, запись «<>» обозначает знак операции «не равно», а запись «<<>» — идущие подряд знаки операций «меньше» и «больше».)

Значением элементарного логического выражения является «истина», если выполняется соответствующее соотношение между значениями арифметических выражений, и «ложь» в противном случае.

Примеры элементарных логических выражений:

$5 < 7$ — имеет значение «истина»;

$(x + 1) <> 2$ — имеет значение «ложь», если значение переменной x есть 1, и значение «истина», если значение переменной x отлично от 1.

Операции *not* («логическое отрицание»), *and* («логическое И»), *or* («логическое ИЛИ») и *xor* («исключающее ИЛИ») задаются следующими таблицами (здесь «И» об-

означает значение «истина», «Л» — «ложь»; операция *not* является одноместной, т. е. имеющей один аргумент):

not:	and: И Л	or: И Л	xor: И Л
И: Л	И: И Л	И: И И	И: Л И
Л: И	Л: Л Л	Л: И Л	Л: И Л

Примеры логических выражений, использующих логические операции (мы предполагаем, что x — переменная вещественного типа):

$\text{not } (x < 3)$ — эквивалентно $x \geq 3$;

$(x < 1.5) \text{ or } (x > 3)$ — принимает значение «истина», если x лежит вне отрезка $[1.5, 3]$;

$(x \geq 1.5) \text{ and } (x \leq 3)$ — принимает значение «истина», если x принадлежит отрезку $[1.5, 3]$.

(Последнее условие — « x принадлежит отрезку $[1.5, 3]$ » — нельзя записать иначе; обычная математическая запись « $1.5 \leq x \leq 3$ » в Паскале недопустима.)

А теперь мы предлагаем вам новый взгляд на выражения в Паскале. Мы говорили о целых арифметических, вещественных арифметических и логических выражениях. Если мы и дальше будем двигаться таким путем, нам потребуются рассмотреть еще несколько видов выражений. Вместо этого удобнее (и логичнее с точки зрения структуры языка) говорить просто о выражениях.

Выражения строятся из констант, переменных и вызовов функций с помощью знаков операций и круглых скобок. Порядок выполнения операций определяется приоритетами операций и скобками (как обычно в математике). Каждая операция требует операндов определенных типов и вырабатывает результат определенного типа (который может отличаться от типов операндов). Некоторые операции допускают несколько вариантов типов операндов; можно считать в таком случае, что имеется несколько операций, которые обозначаются одним и тем же знаком; какая из обозначаемых этим знаком операций должна быть применена, определяется типом операндов. Например, можно считать, что знаком «+» обозначаются четыре операции умножения:

Тип		
левого операнда	правого операнда	результата
целый	целый	целый
целый	вещественный	вещественный
вещественный	целый	вещественный
вещественный	вещественный	целый

С этой точки зрения выражение

$(x + 1) \geq 1.5 \text{ and } (y < 3)$

трактруется следующим образом:

вещественное значение переменной x скла-

дывается с целым значением 1; получается вещественный результат; вещественное значение подвыражения $x+1$ сравнивается с вещественным значением 1.5; получается логический результат; вещественное значение переменной y сравнивается с целым значением 3; получается логический результат;

выполняется операция «логическое И» над логическими значениями подвыражений $(x+1) >= 1.5$ и $(y <= 3)$; получается логический результат.

Приоритеты (1 — высший, 5 — низший) уже известных нам операций задаются следующей таблицей:

Операция	Приоритет	Категория
not	1	Унарные операции - 1
* / div mod and	2	Операции типа умножения
+ - (унарные)	3	Унарные операции - 2
+ - (бинарные) or xor	4	Операции типа сложения
= < > <= >=	5	Операции сравнения

Теперь можно прояснить один момент, который мог показаться загадочным: почему мы в примерах логических выражений вроде

$(x+1) >= 1.5$ and $(y <= 3)$

старательно брали операнды логической операции в скобки (в большинстве других языков программирования этого не требуется). Дело в том, что приоритет операции and выше, чем приоритет операций сравнения, и поэтому по правилам языка в выражении

$x+1 >= 1.5$ and $y <= 3$

подразумевается такая расстановка скобок:

$((x+1) >= (1.5 \text{ and } y)) <= 3$

— а это совсем не то, что нужно, и даже недопустимо, по правилам языка! Поэтому если бы такое выражение было написано без скобок, компилятор выдал бы сообщение об ошибке «типы операндов не соответствуют операции» (так как операция and неприменима к вещественным операндам). По такой диагностике сразу и необразишь, в чем дело, не правда ли? Но компилятор прав, ничего более разумного он сообщить не может (выражение

$(x >= (true \text{ and } y)) <= 3$

в котором x и y — переменные типа boolean, вполне допустимо!).

Чтобы избежать таких загадочных сообщений об ошибках, мы и рекомендуем завести себе правило — всегда заключать операнды логических операций в скобки.

А почему приоритет логических операций сделан таким, что приходится ставить скобки? По-

чему приоритет логических операций не сделан ниже приоритета операций сравнения? Ответ на этот вопрос кроется в деталях языка, которые мы еще не обсуждали, — и не будем обсуждать в настоящем цикле статей. Дело в том, что, кроме операций not, and, or и xor, применимых к операндам логического типа, есть еще операции not, end, or и xor, применимые к операндам целых

типов (они выполняют побитовые операции). Наличием таких операций и объясняется выбор приоритетов.

Теперь вернемся к оператору if. Как мы уже упоминали, в общем случае он имеет формат

```
if <логическое-выражение>
then <оператор1>
else <оператор2> ;
```

Его выполнение происходит следующим образом. Сначала вычисляется логическое выражение. Если его значение есть «истина», то выполняется оператор1, и на этом выполнение оператора if заканчивается. Если же его значение есть «ложь», то выполняется оператор2.

Логическое выражение часто называют условием, а оператор1 и оператор2 — then-альтернативой и else-альтернативой соответственно.

Если в случае ложности условия никаких действий выполнять не нужно, слово else и следующий оператор2 можно опустить. Оператор if в таком случае принимает вид

```
if <логическое-выражение>
then <оператор1> ;
```

Обратите внимание на то, что в случае наличия else-альтернативы «;» после then-альтернативы не ставится. Если вы напишете

```
if x > 0 then a:=1; else a:=-1;
```

то компилятор выдает сообщение об ошибке «Ожидается оператор», и курсор будет установлен на слово else. Действительно, по правилам языка точка с запятой завершает оператор (и компилятор считает, что оператор if не содержит else-альтернативы); после точки должен начинаться новый оператор, а оператор не может начинаться с зарезервированного слова else.

Записывать оператор if мы рекомендуем либо целиком в одной строке программы, либо в несколько строк так, чтобы зарезервированные слова if, then и else начинались в одной и той же позиции (посмотрите, как записан оператор if в примере).

В качестве then- или else-альтернативы оператора может быть указан любой оператор, в том числе и другой оператор if. В такой ситуации может возникнуть неоднозначность. Посмотрите на следующую строку:

```
if <усл1> then if <усл2> then <он1> else <он2>;
```

К какому оператору `if` — первому (внешнему) или второму (вложенному) — относится альтернатива `else`? Для разрешения этой неоднозначности в Паскале принято правило: слово `else` связывается с ближайшим слева словом `if`, с которым еще не связано никакое слово `else`. Таким образом, приведенный выше оператор в соответствии с нашими рекомендациями следует записать так:

```
if <усл1>
then if <усл2>
    then <он1>
    else <он2> ;
```

А как быть, если при выполнении какого-то условия нужно выполнить не один, а несколько операторов? Тут на помощь приходит *составной оператор*. Составной оператор позволяет превратить последовательность операторов в один оператор. Для этого перед первым оператором последовательности нужно поставить зарезервированное слово `begin`, а после последнего — зарезервированное слово `end`. (Зарезервированные слова `begin` и `end` иногда называют «операторными скобками»; заключение последовательности операторов в операторные скобки превращает последовательность в один оператор). Последовательность операторов, расположенная между `begin` и `end`, может содержать произвольное количество операторов, в том числе один или ни одного. Последнее не так уж бессмысленно: в период создания большой программы полезно по ходу дела проверять уже написанные куски. Вы можете размещать пустые операторные скобки `begin` и `end` в тех местах, где в будущем должна появиться последовательность операторов, получая таким способом возможность скомпилировать без ошибок неоконченную программу.

В рассматриваемом примере программы решения квадратного уравнения в операторе `if then-альтернатива` является составным оператором, а `else-альтернатива` — простым.

Использование составного оператора поможет и в том случае, когда слово `else` нужно связать с не с ближайшим слева словом `if`, а с более далеким. Этого можно достичь с помощью такой, например, конструкции:

```
if <усл1>
then begin
    if <усл2>
    then <он1>
    end
else <он2> ;
```

В Паскале имеется и другой условный оператор — `case`. Мы здесь обсудим только простейший вариант его использования; полное описание смотрите в документации.

Рассмотрим пример. Программа должна

ввести с клавиатуры одну из букв А—Д (эта буква понимается как код одной из пяти возможных команд) и выполнить соответствующие действия. Если бы мы использовали оператор `if`, то у нас получилось бы следующее:

```
var c : char;
begin
  c:=ReadKey;
  if c = 'A'
  then begin
    { Действия по команде "А" }
  end
  else if c = 'B'
  then begin
    { Действия по команде "Б" }
  end
  else if c = 'B'
  then begin
    { Действия по команде "В" }
  end
  else if c = 'Г'
  then begin
    { Действия по команде "Г" }
  end
  else if c = 'Д'
  then begin
    { Действия по команде "Д" }
  end
  else writeln
    ('Ошибочная команда');
end.
```

Использование оператора `case` значительно упростит нашу программу:

```
var c : char;
begin
  c:=ReadKey;
  case c of
    'A': begin { Действия по команде "А" } end;
    'Б': begin { Действия по команде "Б" } end;
    'В': begin { Действия по команде "В" } end;
    'Г': begin { Действия по команде "Г" } end;
    'Д': begin { Действия по команде "Д" } end;
    else writeln('Ошибочная команда');
  end(case);
end.
```

Оператор `case` устроен следующим образом. Он начинается с зарезервированного слова `case`; после слова `case` должно стоять выражение (в нашем случае выражение очень простое — оно состоит из одного имени переменной «с»); затем должно стоять зарезервированное слово `of`; дальше идут `case-альтернативы` (в нашем примере их пять; они начинаются с 'А', ..., 'Д'); после них может стоять `else-альтернатива` (ее можно и опустить); заканчивается оператор `case` зарезервированным словом `end`. (Обратите внимание на то, что после слова `end`, завершающего оператор `case`, мы поставили комментарий {case}. Оператор `case` часто бывает длинным и может не помещаться целиком на экране. В таком случае бывает трудно понять, к чему относится слово `end`. Пометка {case} облегчит вам жизнь.) Каждая `case-альтернатива` состоит из константы, следующего за ней двоеточия и оператора, после которого

обязательно должна стоять точка с запятой. Как обычно, если нужно иметь не один оператор, а последовательность, можно воспользоваться составным оператором. Else-альтернатива состоит из зарезервированного слова `else` и следующего за ним оператора.

Заполняется оператор `case` следующим образом. Сначала вычисляется выражение, стоящее между словами `case` и `of`. Затем полученное значение последовательно сравнивается с константами, указанными в начале `case-альтернатив`. Если значение выражения совпадает с какой-либо из констант, то выполняется находящийся в этой альтернативе оператор, и на этом выполнение оператора `case` заканчивается. Если значение выражения не совпало ни с одной из констант, то выполняется оператор, указанный в альтернативе `else`; если же таковая альтернатива отсутствует, то никаких действий не выполняется.

Если для нескольких значений констант должны быть выполнены одни и те же действия, то эти константы могут быть перечислены через запятую в одной альтернативе. Например, если в нашем примере мы хотим, чтобы команды могли вводиться как на верхнем, так и на нижнем регистрах, мы можем написать программу так:

```
var c : char;
begin
  c:=ReadKey;
  case c of
    'a', 'A':begin {действия по команде "A"} end;
    'b', 'B':begin {действия по команде "B"} end;
```

```
'a', 'B':begin {действия по команде "B"} end;
'Г', 'Г':begin {действия по команде "Г"} end;
'b', 'A':begin {действия по команде "A"} end;
else writeln('Ошибка команда');
end(case);
end.
```

И наконец, о небольшой новинке, появившейся в примере программы с оператором `case`.

Мы описали тип переменной «с» как `char`. Это так называемый *литерный тип*. Переменная такого типа занимает в памяти один байт и может принимать одно из 256 значений, которые трактуются как литеры; они могут быть отображены на экране или введены с клавиатуры. Константы этого типа могут записываться двумя способами: во-первых, как литеры, заключенные в апострофы; во-вторых, как знак «≠», за которым следует код литеры. Например, на «Корвете» русская буква «А» имеет код 225; поэтому вместо 'А' мы могли бы написать ≠225 или ≠\$E1 (E1 — шестнадцатеричная запись числа 225). Второй способ удобно использовать для записи управляющих символов, не имеющих графического изображения (возврат каретки, стрелки и т. п.).

`ReadKey` — имя стандартной функции. Она ожидает нажатия клавиши на клавиатуре и после нажатия возвращает в качестве результата введенную литеру.

Продолжение следует.

А. ДЕВЯТОВ

Программирование 9-игольчатых матричных принтеров семейства Epson

Приведем список команд Epson-совместимых принтеров и подробные описания некоторых из них. В списке первый столбец — код команды, второй — ее действие.

Контрольные коды

BEL	Выдача звукового сигнала
BS	Возврат каретки на 1 символ назад
HT	Горизонтальная табуляция
LF	Перевод строки
VT	Вертикальная табуляция
FF	Перемещение бумаги на начало следующей страницы
CR	Печать буфера строки, возврат каретки

SO	Включение двойной ширины символов для одной строки
SI	Включение уплотненного режима печати
DC1	Селектирование принтера
DC2	Отмена уплотненного режима печати
DC3	Деселектирование принтера
DC4	Отмена двойной ширины символов
CAN	Очистка буфера печати (1 строка) от данных
ESC	Контрольный код, с которого начинаются управляющие последовательности кодов
DEL	Удаление из буфера печати последнего символа

Продолжение. Начало см.: Информатика и образование. 1992. № 3—4.

Управляющие ESC-последовательности			
ESC	SO	Включение двойной ширины символов для одной строки	ESC В Установка позиций для вертикальной табуляции
ESC	SI	Включение уплотненного режима печати	ESC С Установка длины страницы в строках
ESC	EM	Подключение устройства для подачи листовой бумаги	ESC CO установка длины страницы в дюймах
ESC	SP	Выбор межзнакового промежутка	ESC D Установка позиций для горизонтальной табуляции
ESC	1	Выбор определенной комбинации шрифтов	ESC E Выбор режима выделенного шрифта
ESC	#	Отмена управления старшим битом данных	ESC F Отмена режима выделенного шрифта
ESC	\$	Установка абсолютной позиции печати	ESC G Установка режима печати «двойной удар»
ESC	%	Переключение между знакогенератором ПЗУ принтера и знакогенератором пользователя	ESC H Отмена режима печати «двойной удар»
ESC	&	Загрузка (определение) знакогенератора пользователя	ESC I Установка режима обработки символов с кодами 0—31 и 128—159 (либо контрольный код, либо печатаемый символ)
ESC	*	Выбор одного из графических режимов	ESC J Печать буфера и перемещенные бумаги на n/216 дюйма
ESC	—	Установка/отмена режима подчеркивания	ESC K Включение графического режима с нормальной плотностью (480 точек на строку)
ESC	/	Выбор канала вертикальной табуляции	ESC L Включение графического режима с двойной плотностью (960 точек на строку)
ESC	0	Установка межстрочного интервала 1/8 дюйма	ESC M Выбор плотности печати «элита» (12 знаков на дюйм)
ESC	1	Установка межстрочного интервала 7/72 дюйма	ESC N Установка количества строк для пропуска поперечной перфорации
ESC	2	Установка межстрочного интервала 1/6 дюйма	ESC O Отмена режима пропуска перфорации
ESC	3	Установка межстрочного интервала n/216 дюйма	ESC P Выбор шага «цицера» (отмена режима ESC M)
ESC	4	Выбор режима курсива (шрифт «italica»)	ESC Q Установка величины правого поля
ESC	5	Отмена режима курсива	ESC R Выбор набора международных символов
ESC	6	Расширение области печатаемого кода (разрешение печати символов с номерами 128—159 и 255)	ESC S Включение режима печати надстрочных/подстрочных символов
ESC	7	Отмена режима ESC 6	ESC T Отмена режима печати надстрочных/подстрочных символов
ESC	8	Отключение датчика конца бумаги	ESC U Включение/отмена печати в одном направлении
ESC	9	Включение датчика конца бумаги	ESC W Включение/отмена режима двойной ширины (печать расширенных символов)
ESC	:	Копирование знакогенератора ПЗУ в ОЗУ принтера	ESC Y Установка графического режима с двойной плотностью и двойной скоростью печати
ESC	<	Выбор однонаправленной печати для одной строки	ESC Z Установка графического режима с учетверенной плотностью печати (1920 точек на строку)
ESC	=	Установка старшего бита (MSB) в 0	ESC Установка относительного положения
ESC	>	Установка старшего бита (MSB) в 1	
ESC	?	Присвоение одному из 8 графических режимов своего символического кода	
ESC	@	Инициализация (сброс принтера)	
ESC	A	Установка межстрочного ин-	

- ESC ^ Установка графического режима печати с использованием всех 9 иглол
- ESC a Установка режима выравнивания текста (по левому, правому краю или центровка)
- ESC b Установка до 16 позиций вертикальной табуляции для одного из каналов
- ESC e Установка инкрементов для горизонтальной и вертикальной табуляций (Compute Mate-160)
- ESC f Печать пробелов или перемещение на строку без возврата каретки (Compute Mate-160)
- ESC i Печать по одному символу (FX-80)
- ESC j Перемещение бумаги в обратном направлении (FX-80)
- ESC k Выбор качественного шрифта
- ESC l Установка величины левого поля
- ESC m Включение/отмена печати символов псевдографики, имеющих коды 128—159 (Compute Mate-160)
- ESC p Установка/отмена режима пропорциональной печати
- ESC s Установка/отмена печати с уменьшенной в 2 раза скоростью
- ESC t Выбор таблицы знаков
- ESC w Установка/отмена режима двойной высоты
- ESC x Установка/отмена режима печати NLQ

Возврат каретки на один символ BS

Формат:

символьный: BS
 десятичный: 8
 шестнадцатеричный: 8
 клавиатурный: <CTRL> H

Печатается буфер строки, после чего печатающая головка возвращается назад на один символ. Из первой позиции строки возврата каретки не происходит. Код BS игнорируется в режиме выравнивания текста («ESC a <2>» или «ESC a <3>»). Если код получен сразу после печати графического образа, то новые данные будут напечатаны на позиции начала печати графического образа.

Возврат каретки CR

Формат:

символьный: CR
 десятичный: 13
 шестнадцатеричный: 0D
 клавиатурный: <CTRL> M

Печать буфера строки и возврат печатающей

головки в первую позицию строки. Перевод строки происходит только в случае установки в соответствующее положение DIP-переключателя или установки линии управления AUTO FEED XT параллельного интерфейса в состояние «НИЗКИЙ УРОВЕНЬ».

Перевод строки LF

Формат:

символьный: LF
 десятичный: 10
 шестнадцатеричный: 0A
 клавиатурный: <CTRL> J

Печать буфера строки и перемещение бумаги на одну строку вверх с учетом текущего значения межстрочного интервала.

Переход к началу следующей страницы FF

Формат:

символьный: FF
 десятичный: 12
 шестнадцатеричный: 0C
 клавиатурный: <CTRL> L

Печать буфера данных и перемещение бумаги на позицию начала следующей страницы с учетом текущего значения длины страницы.

Перевод строки на n/216 дюйма ESC J

Формат:

символьный: ESC J n
 десятичный: 27 74 n
 шестнадцатеричный: 1B 4A n

Печать буфера и перевод бумаги на n/216 дюйма. Величина лежит в диапазоне 0—255. Код не влияет на текущую величину межстрочного интервала и не приводит к возврату каретки.

Очистка буфера печати строки CAN

Формат:

символьный: CAN
 десятичный: 24
 шестнадцатеричный: 18
 клавиатурный: <CTRL> x

Полная очистка буфера печати строки.

Выбор режима печати верхнего/нижнего индекса ESC S

Формат:

символьный: ESC S n
 десятичный: 27 83 n
 шестнадцатеричный: 1B 53 n

n=0 или n=48 — печатается верхний индекс;

n=1 или n=49 — печатается нижний индекс.

В этом режиме все символы имеют высоту, равную 2/3 нормальной, и печатаются под обрез с верхним (n=0) или нижним (n=1) краем строки. В связи с малыми размерами символов печать происходит в режиме «двойного удара».

Отмена режима индекса ESC T

Формат:

символьный: ESC T

десятичный: 27 84
шестнадцатеричный: 1В 54

Отменяется режим печати индексов, установленный командами ESC S 0/1.

Выбор комбинации шрифтов ESC!

Формат:

символьный: ESC ! n
десятичный: 27 33 n
шестнадцатеричный: 1В 21 n

Выбирается комбинация режимов печати, определяемая величиной n (для FX-80 n=0—63, для более поздних модификаций — n=0—255), имеющей побитовую кодировку (каждый бит отвечает за определенный режим печати). Установка бита в 1/0 определяет включение/выключение режима. Значение n определяется суммированием всех требуемых значений в соответствии с табл. 1.

Таблица 1

Режим	Номер бита	Значение
Цицero/элита	0	0/1
Пропорциональный	1	2
Уплотненный	2	4
Выделенный	3	8
«Двойной удар»	4	16
С двойной шириной	5	32
Курсив	6	64
Подчеркивание	7	128

В принтерах FX-80, LX-800/1000, Compute Mate-160 и др. бит 1 не задействован (в принтере FX-80, кроме того, не задействованы биты 6 и 7), следовательно, не включаются и соответствующие режимы. Эти биты при программировании указанных моделей принтеров должны быть установлены в ноль.

Правила совместимости режимов следующие:

в пропорциональном режиме всегда включается выделенный режим;

при выборе шага печати «элита» команды на включение режимов выделенной и уплотненной печати будут проигнорированы;

пропорциональный режим имеет более высокий приоритет, чем уплотненный. При их одновременном выборе команда на включение уплотненной печати не будет выполнена.

В режиме пропорциональной печати каждый знак имеет свою ширину (в отличие от обычной печати, где все знаки имеют постоянную ширину независимо от начертания). Узкий знак при печати занимает меньше места, чем широкий. Таблица ширин знаков в разных принтерах может различаться.

В режиме «двойной удар» каждая строка печатается за два прохода, между которыми бумага продвигается на 1/216 дюйма. Этим достигается «жирное» начертание символов и выделение их на фоне остального текста.

В режиме выделенной печати каждая точка печатается дважды со смещением по горизонтали на 1/120 дюйма.

В режиме подчеркивания все символы, включая пробелы, подчеркиваются.

При плотности печати «цицero» печатается 10 знаков на дюйм, «элита» — 12 знаков на дюйм, при уплотненной печати — 137 знаков в строке длиной 8 дюймов. Во всех трех случаях печать может осуществляться также с удвоенной шириной символов.

Выбор черновиковой/качественной печати ESC x

Формат:

символьный: ESC x n
десятичный: 27 120 n
шестнадцатеричный: 1В 78 n

n=0 или n=48 — включение черновикового шрифта;

n=1 или n=49 — включение качественно-го шрифта (NLQ).

При выборе качественного шрифта печать каждой строки происходит за два прохода печатающей головки с передвижением бумаги на 1/216 дюйма между проходами. Благодаря этому достигается качество печати, близкое к машинописному (Near Letter Quality, сокращенно — NLQ); платой является значительное уменьшение скорости печати. На принтерах FX-800/1000 и LX-800 скорость печати в режиме NLQ составляет 40 и 25 знаков в секунду соответственно.

На принтерах LX-800/1000 и RAVI-8010M режим качественной печати NLQ возможен только для плотности печати «цицero».

Выбор шрифта для качественной печати ESC k

Формат:

символьный: ESC k n
десятичный: 27 107 n
шестнадцатеричный: 1В 6В n

n=0 — выбор шрифта «роман»;

n=1 — выбор шрифта «сансериф».

В режиме качественной печати можно выбрать один из двух вариантов начертания символов с указанными выше названиями. Знаки шрифта «роман» более высокие и округленные, похожи на шрифт портативной пишущей машинки. Шрифт «сансериф» более строг, прямоуголен.

В принтерах RAVI-8010M и Compute Mate-160 эта команда не действует, так как у них есть только один качественный шрифт. Управление расстоянием между строк.

Минимальное расстояние, на которое можно продвинуть бумагу, равно 1/216 дюйма, т. е. около 0,118 мм. Соответствие между общепринятым в машинописи понятием интервала и дюймовым стандартом следующее:
1 интервал=1/6 дюйма (36/216 дюйма);
1,5 интервала=1/4 дюйма (54/216 дюйма);
2 интервала=1/3 дюйма (72/216 дюйма).

Существуют пять команд для установки величины интервала между строками; они приведены в табл. 2.

шестнадцатеричный: 1B 43 n
Установка длины страницы, равной n строк, n=1—127. Эта величина при последующих

Таблица 2

Символьный	Код		Величина межстрочного интервала
	Десятичный	Шестнадцатеричный	
ESC 0	27 48	1B 30	1/8 дюйма
ESC 1	27 49	1B 31	7/72 дюйма
ESC 2	27 50	1B 32	1/6 дюйма
ESC 3 n	27 51 n	1B 33 n	n/216 дюйма (n от 0 до 255)
ESC A n	27 65 n	1B 41 n	n/72 дюйма (n от 0 до 85)

Для команды ESC 3 n при n=1 или n=2 фирма-производитель не гарантирует точность выполнения перевода строки. (Хотя автору по опыту работы известно о практически стопроцентной надежности выполнения перевода строки на 1/216 дюйма.)
Установка величины левого поля ESC 1

Формат:

символьный: ESC 1 n
десятичный: 27 108 n
шестнадцатеричный: 1B 6C n

Величина левого поля устанавливается равной n ширин символов, используемых в данный момент, и не изменяется при изменении плотности печати. Установка в пропорциональном режиме осуществляется, как в режиме «цицero».

Максимальная величина левого поля (в знаках) определяется текущей плотностью печати:

режим «цицero» n < 79;
режим «элита» n < 94;
уплотненный режим n < 135

(для принтера с шириной валика 8 дюймов).

В режиме печати расширенных символов переменная n не должна превышать половины указанных значений.

Установка величины правого поля ESC Q

Формат:

символьный: ESC Q n
десятичный: 27 81 n
шестнадцатеричный: 1B 51 n

Аналогична установке левого поля. Разрешенные диапазоны для переменной n для принтера с шириной валика 8 дюймов:

режим «цицero» n=2—80;
режим «элита» n=3—96;
уплотненный режим n=4—137.

Минимальный интервал между величинами правого и левого полей соответствует двум знакам шрифта «цицero».

Установка длины страницы в строках ESC C

Формат:

символьный: ESC C n
десятичный: 27 67 n

изменениях межстрочного интервала не изменяется.

Команды FF (переход к новой странице) и ESC N (пропуск поперечной перфорации) выполняются в соответствии с текущим значением длины страницы.

Установка длины страницы в дюймах ESC C 0

Формат:

символьный: ESC C NUL n
десятичный: 27 67 0 n
шестнадцатеричный: 1B 43 0 n

Установка длины страницы в дюймах, n=1—22. Команды перехода к новой странице и пропуска поперечной перфорации выполняются в соответствии с текущим значением длины страницы.

Установка абсолютной позиции печати ESC \$

Формат:

символьный: ESC \$ n1 n2
десятичный: 27 36 n1 n2
шестнадцатеричный: 1B 24 n1 n2

Задание позиции для последующей печати. Позиция определяется, как расстояние в $(n1 + (n2 \times 256)) / 60$ дюймов от текущей позиции левого поля. При выходе за правое поле команда не выполняется.

У FX-80 и Compute Mate-160 эта команда отсутствует.

Выравнивание текста в процессе печати ESC a

Формат:

символьный: ESC a n
десятичный: 27 97 n
шестнадцатеричный: 1B 61 n

Работает только в режиме качественной печати NLQ. Возможны следующие значения n:

- 0 — выравнивание текста по левому полю;
- 1 — центрирование текста;
- 2 — выравнивание текста по правому полю;
- 3 — полное выравнивание текста.

При включении питания выбирается режим, соответствующий n=0. Полное выравнивание (n=3) выполняется при заполнении буфера печати или при получении кодов CR,

VT, LF, FF. При $n=3$ в пределах параграфа не должен передаваться код возврата каретки. Команды горизонтальной табуляции HT и возврата каретки на один символ BS работают только в режиме $n=0$.

У FX-80 и RAVI-8010M эта команда отсутствует.

Установка режима курсива ESC 4

Выбирается таблица символов со шрифтом «курсив». На аппаратно русифицированных принтерах (т. е. со знакогенератором кириллицы, «зашитым» в ПЗУ) курсивом печатаются только знаки английского алфавита.

Печать курсивом производится на уменьшенной в два раза скорости.

Выбор таблицы символов ESC t

Формат:

символьный:	ESC	t	n
десятичный:	27	116	n
шестнадцатеричный:	1B	74	n

Выбор таблицы символов для знаков с кодами 128—255. При $n=0$ выбирается таблица курсива, при $n=1$ — таблица с графическими символами (сами графические символы в ней имеют коды с номерами 176—223).

У FX-80 и Compute Mate-160 эта команда отсутствует.

Расширение области печатаемых символов ESC I

Формат:

символьный:	ESC	I	n
десятичный:	27	73	n
шестнадцатеричный:	1B	49	n

При включении принтера или после его инициализации символы с десятичными кодами 0—31, 128—159 рассматриваются им как управляющие. Они становятся печатаемыми при $n=1$ и контрольными при $n=0$.

Команда ESC I 1 позволяет использовать указанные коды при программировании пользователем собственного знакогенератора. Копирование таблицы символов из ПЗУ в ОЗУ ESC :

Формат:

символьный:	ESC	:	NUL	NUL	NUL
десятичный:	27	58	0	0	0
шестнадцатеричный:	1B	3A	0	0	0

Копирует всю таблицу символов (коды 0—255) из постоянной в оперативную память принтера. Это необходимо в тех случаях, когда пользователь собирается дополнить существующую таблицу ASCII-символов собственными символами.

Первая команда трехступенчатого процесса определения пользователем собственного набора символов.

Выбор набора символов ПЗУ/ОЗУ ESC %

Формат:

символьный:	ESC	%	n
десятичный:	27	37	n

шестнадцатеричный: 1B 25 n

$n=0$ — таблица символов ПЗУ;

$n=1$ — таблица символов ОЗУ.

Переключение между знакогенератором, постоянно хранящимся в ПЗУ, и тем, который записан в данный момент в ОЗУ.

Команда обычно выполняется вслед за командой копирования знакогенератора ESC: 0 0 0 в процессе определения пользователем собственных знаков.

Загрузка внешнего знакогенератора ESC & Формат

символьный: ESC & NUL n m a d1... d1e1... e1f1...

десятичный: 27 37 0 n m a d1... d1e1... e1f1...

шестнадцатеричный: 1B 25 0 n m a d1... d1e1... e1f1...

n — код начального символа для программирования;

m — код конечного символа для программирования;

a — управляющая константа;

d1...d1 — описание матрицы первого перепрограммируемого знака;

e1...e1 — описание матрицы второго перепрограммируемого знака и т. д.

Перепрограммирует символы в знакогенераторе, находящемся в ОЗУ принтера.

Основные сведения о программировании собственного набора знаков приведены в статье А. Дмитриева в «ИНФО» № 2 и в данной публикации опущены.

Выбор графического режима с одинарной плотностью ESC K

Формат

символьный: ESC K n1 n2 d1... dk

десятичный: 27 75 n1 n2 d1... dk

шестнадцатеричный: 1B 48 n1 n2 d1... dk

Выбирается режим графической печати с одинарной плотностью, равной 60 точкам на дюйм (максимальное число точек на строке принтера с шириной зоны печати 8 дюймов равно 480). С такой плотностью печатаются символы шрифтов «цицера» и «элита».

В графических режимах пользователь сам управляет каждой из 8 игл печатающей головки. (Работа со всеми 9 иглами возможна с помощью команды ESC\.) Числа d1...dk несут в себе информацию о графическом образе, состоящем из k вертикальных колонок. Каждое число d в последовательности d1...dk содержит информацию о том, ударять или нет той или иной иглой в данной вертикальной колонке. Число d имеет побитную кодировку, и его значение вычисляется исходя из табл. 3.

При необходимости ударить той или иной иглой в данной вертикальной колонке нужно просуммировать соответствующие этим иглам коды; результат суммирования даст величину d.

Таблица 3

Номер иглы	Код	Положение в головке
1	128	Самая верхняя игла
2	64	
3	32	
4	16	
5	8	
6	4	
7	2	
8	1	Вторая снизу игла

Числа $n1$ и $n2$ указывают общее число вертикальных колонок в графическом образе. Они вычисляются по следующему правилу: $n2$ есть результат целочисленного деления числа данных на 256, а $n1$ — остаток от такого деления:

$$n1 = \langle \text{число колонок данных} \rangle \text{ MOD } 256,$$

$$n2 = \langle \text{число колонок данных} \rangle \text{ DIV } 256.$$

Например, если число колонок равно 400, то $n2=1$, $n1=144$.

При передаче графической команды могут быть допущены следующие ошибки:

данных не хватает, т. е. число элементов в последовательности $d1...dk$ меньше, чем $n2 \times 256 + n1$. В этом случае принтер остановится и будет ждать дополнительных данных;

переданы избыточные данные, т. е. число элементов в последовательности $d1...dk$ больше, чем $n2 \times 256 + n1$. В этом случае принтер сначала напечатает графический образ, затем вернется в текстовый режим и напечатает избыточную часть данных в виде текста, интерпретируя данные d как коды печатаемых символов.

Если число $(n2 \times 256) + n1$ превышает предельно допустимое число колонок для данного графического режима, т. е. образ выходит за правый край поля печати, то лишние данные просто не будут напечатаны.

При печати на одной строке текста и графических образов надо учитывать ширину печатаемых символов. В табл. 4 приводятся данные о ширине (в точках одинарной плотности, равных 1/60 дюйма) одного символа в различных режимах печати.

Таблица 4

Режим печати	Число точек
«Цицеро», «выделенный»	6
«Элита»	5
«Уплотненный»	3,5

Выбор графического режима с двойной плотностью ESC L

Формат:

символьный: ESC L $n1 n2 d1...dk$

десятичный: 27 76 $n1 n2 d1...dk$

шестнадцатеричный: 1B 4C $n1 n2 d1...dk$

Выбирается режим графической печати с двойной плотностью, равной 120 точкам на дюйм. Правила формирования $n1$, $n2$ и последовательности $d1...dk$ аналогичны используемым в команде ESC K.

Печать производится с уменьшенной вдвое по сравнению с режимом «ESC K» скоростью.

Выбор графического режима с учетверенной плотностью ESC Z

Формат:

символьный: ESC Z $n1 n2 d1...dk$

десятичный: 27 90 $n1 n2 d1...dk$

шестнадцатеричный: 1B 5A $n1 n2 d1...dk$

Выбирается режим графической печати с учетверенной плотностью, равной 240 точкам на дюйм. Запрещается печать соседних по горизонтали точек. Правила формирования $n1$, $n2$ и последовательности $d1...dk$ аналогичны используемым в команде ESC K.

Печать производится с уменьшенной вдвое по сравнению с режимом «ESC K» скоростью. Сброс принтера ESC @

Формат:

символьный: ESC @

десятичный: 27 64

шестнадцатеричный: 1B 40

Сброс состояния принтера и очистка буфера строки. Принтер приходит в состояние, аналогичное состоянию после включения питания.

Звуковой сигнал BEL

Формат:

символьный: BEL

десятичный: 7

шестнадцатеричный: 7

клавиатурный: <CTRL> C

Принтер выдает звуковой сигнал, тип и громкость которого зависят от типа принтера.

Отключение датчика конца бумаги ESC 8

Формат:

символьный: ESC 8

десятичный: 27 56

шестнадцатеричный: 1B 38

Отключается датчик конца бумаги. Команда не оказывает влияния на функционирование сигнального светодиода PAPER OUT на верхней панели управления принтера.

Используется для печати до конца листовой бумаги.

Включение датчика конца бумаги ESC 9

Формат:

символьный: ESC 9

десятичный: 27 57

шестнадцатеричный: 1B 39

Включается датчик конца бумаги.

При достижении позиции печати, отстоящей примерно на 3/4 дюйма от нижнего края листа бумаги, принтер переходит в автономный режим, что сопровождается погасанием индикатора ON LINE, зажиганием красного индикатора PAPER OUT на панели управления принтера и выдачей звукового сигнала.

Печать одной строки в одном направлении ESC <

Формат:

символьный:	ESC <
десятичный:	27 60
шестнадцатеричный:	1B 3C

Печатающая головка возвращается в крайнюю левую позицию, после чего текущая строка из буфера печатается слева направо.

Режим используется для уменьшения гори-

зонтальных искажений, всегда существующих из-за неидеальности печатающего механизма.

Печать в одном направлении ESC U

Формат:

символьный:	ESC U n
десятичный:	27 85 n
шестнадцатеричный:	1B 55 n

n=0 или n=48 — выбор двунаправленной печати;

n=1 или n=49 — выбор однонаправленной печати.

Однонаправленный режим позволяет достичь более точного позиционирования печатающей головки относительно бумаги

Окончание следует.

С. ГРИГОРЬЕВ

53

Конфигурация для Пролога-Д

Пролог-Д — это система логического программирования, ориентированная на применение в образовании. Она реализована на учебных машинах [1]. Программы, разработанные с ее помощью, используются в учебном процессе [2, 3]. Однако попытки создания сложных программ на входном языке системы Пролог-Д ограничены техническими ресурсами учебных ПЭВМ. Одним из способов преодоления этих ограничений может быть разработка специальных программных средств — конфигураторов, адаптирующих интерпретатор Пролога-Д к разрабатываемой программе и к имеющимся техническим средствам ПЭВМ. В литературе описано использование конфигураторов для встроенных в технические устройства программных систем управления, написанных на процедурных языках [4]. В данной работе рассматриваются способы построения конфигулятора для системы логического программирования и приводятся примеры адаптации системы Пролог-Д для учебных применений.

Программа на языке Пролог-Д рассматривается совместно с интерпретатором. Это объединение включает все необходимые компоненты для исполнения программы на вычислительной машине. Интерпретатор Пролог-Д является универсальной программой, включающей в себя сегменты, не требуемые при выполнении данной программы, например некоторые встроенные предикаты, редактор исходного текста и др. Потребность в них определяется только из анализа текста программы. В процессе работы программы,

например при вводе данных, могут измениться значения переменных, которые в свою очередь изменят структуру программы. Структура программы может измениться и за счет конкретизации переменных вопроса.

Алгоритм построения конфигулятора состоит из двух разделов: оптимизации текста исходной программы и оптимизации программы интерпретатора.

Первый раздел включает две стадии:

1. Удаление из исходного текста программы всех предложений, имена головы которых не являются именами подцели вопроса или могут быть получены из подцели вопроса с помощью процедуры вывода Пролога.

2. Все переменные, фиксированные в вопросе постоянными, подставляются во все предложения, для которых вопрос есть родительское предложение. Если подцель некоторых предложений оказывается невыполненной, то эти предложения удаляются из текста программы. Если подцели выполнены, то они могут быть удалены из предложения.

Оптимизация исходного текста программы поясняется примером.

Предположим, что программа на Прологе имеет вид:

```
PO(x,y,z) <- A1(x), A2(y,z);
PO(x,y,z) <- R1(x), R2(y,z);
PO(x,y,z) <- B1(z), B2(y,z);
A1(atom);
R1(atom) <- ЛОЖЬ;
Q(x,y,z,t) <- L1(x,t), L2(y,z);
?PO(atom,y,z);
```

После обработки на первой стадии получается:

$PO(x,y,z) \leftarrow A1(x), A2(y,z);$

$PO(x,y,z) \leftarrow R1(x), R2(y,z);$

$PO(x,y,z) \leftarrow B1(z), B2(y,z);$

$A1(atom);$

$R1(atom) \leftarrow \text{ЛОЖЬ};$

$?PO(atom,y,z);$

После обработки на второй стадии получается:

$PO(atom,y,z) \leftarrow A2(y,z);$

$PO(atom,y,z) \leftarrow B1(z), B2(y,z);$

$?PO(atom,y,z);$

Второй этап — оптимизация интерпретатора осуществляется за счет удаления фрагментов системы, не нужных для решения данной конкретной задачи. К таким фрагментам относятся блок встроенных предикатов (удаляются те разделы блока, которые содержат описание предикатов, не встречаю-

щихся в программе), редактор исходного текста. Для системы Пролог-Д это возможно, поскольку она является открытой для пользователя и допускает изменения и на уровне набора встроенных предикатов, и на уровне алгоритмов. Полное описание системы и ее алгоритмов приведено в [4].

Литература

1. Григорьев С. Г. Учебные трансляторы с языка Пролог // Труды VI Всесоюзного семинара «Разработка и применение программных средств ПЭВМ в учебном процессе»: Тезисы докладов. М.: ИПИ АН СССР, 1991.

2. Федюшин Д. П. Парадигмы программирования // Информатика и образование. 1991. № 5.

3. Фаулджер Р. Программирование встроенных микропроцессоров. М.: Мир, 1985.

4. Григорьев С. Г. Реализация систем логического программирования для ПЭВМ с ограниченными ресурсами и ее применения в образовании, медицине и технике. М.: Школа-Пресс, 1992.

ОРИГИНАЛЬНАЯ СИСТЕМА ПРОГРАММИРОВАНИЯ ДЛЯ ИСПОЛЬЗОВАНИЯ В УЧЕБНОМ ПРОЦЕССЕ

К — система

простота освоения и использования
уникальный механизм исключений из правил
современный математический аппарат
лицензионная чистота
запись правил на РУССКОМ языке
доступ к наиболее распространенным базам данных

К — система дает возможность организовать учебный процесс по информатике, программированию, биологии, истории, физике и другим дисциплинам на самом современном уровне.

Мы бесплатно предлагаем Вам использовать наши методические разработки в следующих курсах:

Курс «Основы программирования» (32 часа), ориентированный на школьников среднего и старшего возраста, а также студентов, впервые приступающих к изучению информатики.

Курс «Логическое программирование» (96 часов), ориентированный на школьников старших классов, студентов младших курсов. Цель курса — познакомить с основами логического вывода, дать практические навыки по разработке программ и баз знаний.

Курс «Представление знаний в продукционно-логических системах» (256 часов), ориентированный на студентов старших курсов и аспирантов, нацелен на подготовку специалистов в области искусственного интеллекта и баз знаний.

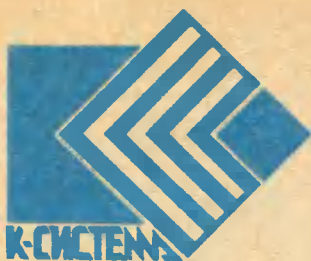
К — система реализована для персональных компьютеров типа IBM PC в среде MS DOS.

Наши специалисты готовы провести консультации по рациональному использованию К — системы в обучении.

Контактные телефоны: 362-89-87, 921-57-51.

Адрес для переписки: 129164, Москва, а/я 46.

Мы рады сотрудничать с Вами!



К — система

Если сравнить принципы программирования современных ЭВМ и методы обучения человека, то обнаруживается существенное различие в степени динамичности этих процессов.

Традиционное программирование выполняется по схеме: постановка задачи → алгоритм → программа.

Внесение качественных изменений в уже разработанную программу, как правило, сопряжено со значительными трудозатратами и зачастую приводит к необходимости перепрограммирования задачи заново.

Таким образом, традиционное программирование носит преимущественно статический характер. В противоположность этому обучение (или «программирование») человека происходит непрерывно в течение десятков лет без какого-либо заранее определенного плана или «алгоритма». Обучение — это существенно динамический процесс. Именно на подобный динамический стиль программирования и ориентирована К — система. В К — системе программирование выполняется по естественной итеративной схеме:

задача → программа → эксплуатация

К — система реализует новый принцип программирования, отличающийся автоматическим распознаванием и интерпретацией исключений из правил. В К — системе возможно программирование даже таких задач, алгоритм решения которых не известен программисту. В этом смысле можно говорить о программировании неформальных процедур.

Математические основы К — систем наиболее полно изложены в монографии Кузнецова В. Е. «Представление в ЭВМ неформальных процедур». М.: Наука. 1989. Серия «Искусственный интеллект».

В математическом плане К — система — это формальная система (типа исчисления Поста) с рекурсивным отношением «исключения» на множестве выводов.

В настоящее время К — система реализована программно на IBM PC XT/AT.

В качестве языка программирования в К — системе используется язык К, который дает возможность в доступной форме и с единых позиций познакомить учащихся с основными направлениями современных информационных технологий (основы программирования, представление знаний, базы данных и знаний, логический вывод, немонотонный вывод, рекурсия, режим возвратов). Язык К выгодно отличается от Пролога простотой синтаксиса и функциональностью. Основная особенность языка К — наличие встроенного механизма исключений из правил*.

Базовый комплект поставки К — системы версии 2.10 включает:

Диск 1 (360 Кб): модуль среды разработки К — программ; служебные файлы конфигурации, сообщений об ошибках, помощи; исполнитель готовых К — программ.

Диск 2 (360 Кб): транслятор из текстового вида во внутреннее представление; модули подсистемы расширения пользователя.

Диск 3 (360 Кб): обучающая программа по языку К.

Диск 4 (360 Кб): демонстрационные К — программы.

Дополнительно поставляются модули, обеспечивающие доступ к данным dBASE (2 диска по 360 Кб) или CLIPPER (2 диска по 360 Кб).

Прилагается документация в виде брошюры объемом около 100 страниц.

Новая версия 3.0 позволяет пользователям встраивать механизм логического вывода в готовый программный продукт. Комплект поставки содержит исходные тексты К — системы на языке Си, совместимые с Microsoft C и TURBO C.

Зарегистрированные пользователи имеют возможность приобретать новые версии К — системы на льготных условиях. Предоставляется 50 % скидка учебным заведениям.

* В первом номере журнала «Информатика и образование» в 1993 г. будет опубликована статья д-ра техн. наук В. Е. Кузнецова о языке программирования К.

Стандарт MSX для записи на МЛ

Важнейшим внешним устройством для хранения информации в большинстве отечественных бытовых компьютеров пока остается магнитофон. В связи с этим определенный интерес для пользователей представляют сведения о способах и форматах записи на магнитную ленту (МЛ).

Существует несколько методов кодирования цифровой информации на МЛ. Так, широкоизвестный домашний компьютер БК различает единицу и ноль по длительности считываемого импульса. Многие компьютеры («Радио-86ПК», «Микроша», «Специалист» и др.) реагируют на перепады напряжения: скачок от высокого потенциала к низкому воспринимается как ноль, а противоположный — как единица. Наконец, в последнее время все большее распространение получает стандарт MSX, в основу которого положено различие частоты сигналов нуля и единицы.

Стандарт MSX разработан в 1984 г. под руководством молодого японского предпринимателя К. Ниси. Предполагалось, что наличие этого стандарта приведет к полной совместимости бытовой вычислительной техники разных производителей подобно тому, как телевизионный стандарт позволяет нам без особых проблем смотреть передачи с помощью телевизионных приемников, имеющих заметно различающееся внутреннее устройство. В силу ряда экономических причин стандарт MSX не стал всеобщим, хотя и оказал заметное влияние на развитие производства компьютеров. Несколько миллионов MSX-машин распространились по Японии, Южной Корее, Голландии, Испании, Италии, ФРГ. Созданы ассоциации пользователей MSX, издается множество специализированных журналов и бюллетеней новостей.

Причиной популярности стандарта MSX в Советском Союзе, без сомнения, послужила массовая закупка КУВТ «Ямаха». Благодаря широкому возможностям и высокой надежности «Ямахи» быстро завоевали симпатии советских пользователей. Не удивительно, что они оказали существенное влияние на разработку новых отечественных компьютеров. Появились такие «ямахоподобные» ПЭВМ (подобие в основном касается языка Бейсик и форматов представления информации), как «Корвет», «Форманта», а также несколько разновидностей модели ПК-8000 — «Сура», «Веста» и др. Даже в далеком от MSX компьютере «Ириша» был принят совместимый формат записи на МЛ.

ПК-8000 — модель, наиболее близкая к MSX-компьютеру «Ямаха». Сделана попытка не просто реализовать совместимый с MSX-стандартом Бейсик на отечественной ЭВМ, но по возможности повторить архитектуру «Ямахи MSX-1», режимы работы ее дисплея и других внешних устройств. Конечно, отечественная элементная база не позволила реализовать все возможности MSX-1, особенно в области видеоэффектов (спрайтов и т. п.). Но определенная MSX-совместимость все же была достигнута.

Страничный (по 16 Кбайт) способ организации памяти на ПК-8000 совпадает с принятым в «Ямахе»; выбор нужной конфигурации памяти в обеих машинах задается одинаково (содержимым специального порта). Предусмотрена возможность расширения памяти. Реализованы режимы экрана SCREEN 0—2, причем достаточно близко к «оригиналу». Достигнута полная совместимость на уровне магнитофона (кассета, записанная на «Ямахе», может быть прочитана на ПК-8000, и наоборот).

Встроенный Бейсик-интерпретатор имеет достаточно хорошую совместимость с имеющимся в «Ямахе», если не учитывать не реализованные из-за аппаратных трудностей спрайты и звуковые эффекты. К сожалению, в версии 1.1 для ПК-8000 заявлены, но не реализованы графические операторы DRAW, CIRCLE и PAINT, хотя технические возможности их исполнения имеются. По-видимому, этот недостаток объясняется стремлением разместить интерпретатор на одной странице ПЗУ (16 Кбайт; в MSX-1 интерпретатор вдвое больше!). Очень хотелось бы, чтобы появилась новая версия Бейсика, в которой заявленные операторы все-таки заработают.

ПК-8000 выгодно отличается от всех отечественных ПЭВМ наличием полноэкранный редактора Бейсик-программы. Например, вы можете подвести курсор к нужной строке, исправить ее и нажать клавишу ввода.

К сожалению, имеется, и довольно много, мелких отличий типа отсутствия конструкции ELSE. Кроме того, ПК-8000 собран на процессоре K580, возможности которого значительно меньше, чем у стоящего в «Ямахе» Z80. И все же пользователи ПЭВМ типа «Веста» (в частности, автор этих строк) за умеренную стоимость приобретают неплохой бытовой компьютер для серьезной работы, а не просто игровой автомат.

Существует дисковый вариант ПЭВМ «Сура» с ОС типа CP/M. Он очень мало распространен, и информации о нем также немного.

Рассмотрим основы магнитофонной части стандарта MSX. Она регламентирует способ записи на МЛ единичного и нулевого бита, формат представления отдельного байта, а также группировку информации в блоки, снабженные специальными маркерами-заголовками.

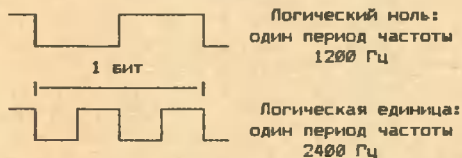
Каждый бит записывается на МЛ методом двухчастотного кодирования FSK (Frequency

Shift Keying, известен также как стандарт «Kansas City»), логическому нулю соответствует один период частоты 1200 Гц, единице — два периода частоты 2400 Гц. Общая длительность нуля и единицы, таким образом, является одинаковой (рис. 1).



При считывании определяется число перепадов напряжения за определенное время (обычно это 3/4 периода сигнала 1200 Гц): отсутствие изменений или только один перепад в течение этого времени принимается за ноль, два или три — за единицу; при регистрации большего количества перепадов фиксируется ошибка.

Каждому байту на МЛ предшествует дополнительный нулевой бит (стартовый), после записи байта формируются два единичных бита (стоповых) (рис. 2).



MSX-стандарт предусматривает разделение данных на блоки, каждому из которых предшествует «короткий» заголовок — 4096 периодов сигнала частотой 2400 Гц. Перед началом любого файла помещается отдельный «длинный» заголовок (16 384 периода той же частоты) и информация о типе и имени файла. Заголовок не только идентифицирует начало файла, но и позволяет подстроиться под скорость движения ленты на используемом магнитофоне. Для этого программа чтения заголовка измеряет время прохождения фиксированного числа импульсов (обычно 256) и определяет среднее время одного импульса. С использованием полученной величины вычисляются требуемые для чтения константы; их значения заносятся в специальные ячейки служебной области.

Описываемая структура файла на МЛ хорошо контролируется «на слух», в чем читатели, вероятно, уже убедились при «воспроизведении» файлов.

В MSX-Бейсике используются файлы трех типов [1], каждый из которых имеет свои особенности записи на МЛ. Рассмотрим их подробнее.

1. Программа на Бейсике в текстовом формате, записанная по команде SAVE; фактически это выполнение команды LIST

на МЛ. Для краткости такое представление часто называют ASCII-форматом. Текст автоматически разбивается на отдельные блоки по 256 байтов, в последнем помещается стандартный код конца текстового файла 1A. Это позволяет игнорировать «лишнюю» информацию в последнем блоке, который физически имеет полную длину 256 байтов.

В случае работы с файлом последовательного доступа информации на МЛ он также записывается в ASCII-формате.

2. Программа на Бейсике во внутреннем представлении, получающаяся на МЛ после выполнения команды CSAVE. В этом случае файл представляет собой копию области ОЗУ, в которой хранится программа (известно, что текст программы для хранения преобразуется, что ускоряет его расшифровку при исполнении и уменьшает занимаемый объем памяти). Информация в таком формате на блоки не разбивается; конец файла определяют 11 нулей, идущих подряд.

Достоинством этого режима записи является существенная экономия времени и места на МЛ, недостатком — невозможность чтения такой записи на компьютере другой модели из-за несовпадения внутреннего представления программ.

3. Запись на МЛ содержимого памяти в двоичном виде по команде BSAVE. Чаще всего это программа в машинных кодах, хотя так могут быть записаны и данные. На МЛ формируется единый блок, перед ним автоматически вставляются три двухбайтовых числа: адреса начала, конца и запуска файла (те адреса, которые вы указываете в качестве параметров в команде BSAVE после имени файла).

Чтобы понять, как компьютер различает эти типы файлов, необходимо вернуться к строению «длинного» заголовка. Помимо настроенной последовательности сигналов частотой 2400 Гц он содержит 16 служебных байтов, последние 6 из которых — имя файла. Первые 10 байтов одинаковые и определяют тип файла: 10 кодов EA — файл в ASCII-формате, коды D0 — двоичный файл, D3 — файл с внутренним представлением программы.

Полезность приведенных сведений продемонстрируем на примере одной особенности работы Бейсик-интерпретатора с магнитофоном. Вы, наверное, замечали, что по командам LOAD, CLOAD и BLOAD компьютер реагирует только на заголовки заданного типа; все остальные файлы полностью игнорируются, как будто их вовсе нет на МЛ. Это не всегда удобно, особенно если на магнитофоне стоит кассета с неизвестным заранее содержимым. Короткая про-

грамма чтения заголовка позволяет легко устранить этот недостаток. Ниже приводится работоспособный фрагмент ассемблерной программы для ПЭВМ «Ямаха», демонстрирующий принципы организации самостоятельной обработки информации заголовка на МЛ. Работа программы напоминает выполнение команды FILES: после чтения заголовка на экран выводится тип (A — ASCII; B — BIN, т. е. двоичный; C — COD — внутреннее представление) и имя файла.

BEGIN:	9000 CD E9 72	CALL MARKER:72E9	; Чтение маркера
	9003 06 09	LD B,09	; Число байтов - 1
	9005 CD D4 72	CALL RDBYTE:72D4	; Чтение байта с МЛ
	9008 4F	LD C,A	; Запомнить 1-й байт
L1:	9009 CD D4 72	CALL RDBYTE:72D4	; Чтение байта с МЛ
	900C B9	CP C	; Сравнить с 1-м байтом
	900D 20 F1	JR NZ,BEGIN:9000	; На начало при несопадении
	900F 10 F8	DJNZ L1:9009	; К проверке следующего байта
	9011 21 71 F8	LD HL,F871	; Адрес буфера имени файла
	9014 06 06	LD B,06	; Длина имени
L2:	9016 CD D4 72	CALL RDBYTE:72D4	; Чтение байта с МЛ
	9019 77	LD (HL),A	; Запись очередного байта
	901A 23	INC HL	; Переход к чтению
	901B 10 F9	DJNZ L2:9016	; следующего байта имени
TYPE:	901D 79	LD A,C	; Тип файла (EA, D0, D3)
	901E CB 67	BIT 4,A	; Перевод кода:
	9020 28 02	JR Z,L3:9024	; EA=>41 (4-й бит во 2-й,
	9022 CB D7	SET 2,A	; D0=>42 сдвиг вправо и сврос
L3:	9024 0F	RRCA	; D3=>43 лишних битов)
	9025 E6 43	AND 43	;
PRINT:	9027 DF	RST 18	; Вывод полученного символа
	9028 3E 20	LD A,20	; Вывод
	902A DF	RST 18	; провала
	902B CD 18 71	CALL PRINAM:7118	; Вывод имени из буфера
	902E 18 D0	JR BEGIN:9000	; На начало

Подробные комментарии позволяют легко разобраться в приведенном фрагменте. Отметим только, что выход из программы происходит по нажатию комбинации клавиш CTRL и STOP, анализ которой предусмотрен внутри используемых «магнитофонных» подпрограмм Бейсик-интерпретатора.

Используя приводимые в тексте примера коды команд процессора, читатели могут сформировать BIN-файл, даже не разбираясь в ассемблере. Те же, кто знаком с этим языком, легко могут усовершенствовать программу: сделать более красивый вывод на экран, добавить чтение адресов у BIN-файлов и т. д.

Программа является переносимой и может быть загружена с любого адреса.

ДОСКА ОБЪЯВЛЕНИЙ

Предлагаю переписываться и обмениваться программами, опытом работающим на УКНЦ (МС 0511), НГМД МС 5309. 346302, Ростовская обл., Каменский р-н, пос. Богдановка, средняя школа. А. Б. Мионов.

Аналогичная программа легко реализуется и на перечисленных выше отечественных ПЭВМ с MSX-форматом записи на МЛ. Автор, например, давно и успешно использует несколько расширенную программу такого типа на бытовой ЭВМ «Веста».

Читателям, пожелавшим самостоятельно реализовать MSX-стандарт на своих компьютерах с процессором K580, рекомендуем обратиться к подробной статье [2].

В заключение — замечание техниче-

ского характера. Некоторые пользователи испытывают трудности в подключении магнитофона к «Ямахе» из-за несоответствия стандартов на разъемы. Но проблема эта кажущаяся! Несмотря на большое число гнезд в магнитофонной розетке «Ямахи», для подключения можно использовать стандартную отечественную пятиштырьковую вилку. Разводка сигналов такова:

- 2 — общий провод («земля»);
- 4 — к микрофонному входу магнитофона;
- 5 — к линейному выходу магнитофона.

Литература

1. Блэнд Г. Основы программирования на языке Бейсик в стандарте MSX. М.: Финансы и статистика, 1989.
2. Барышников Б. Н. и др. // Микропроцессорные средства и системы. 1986. № 3. С. 59—64.

Хочу переписываться и обмениваться программами с владельцами ПЭВМ «Привец-8Д».

292040, Львовская обл., Жовкивский р-н, г. Дубляны, ул. Студенческая, 1/3. В. В. Здорюк.

Имею работающий дисковод на БК-0010.01, приглашаю единомышленников, работающих с дисководами, к переписке, обмену программами, идеями. 612220, Кировская обл., г. Яранск, а/я 2. С. П. Таланов.

Прошу помочь в приобретении программы, методической литературы, дешевых джойстиков с контроллерами для класса УКНЦ. Хотел бы переписываться с владельцами УКНЦ (МС 0511). 453630, Башкирия, Абзелиловский р-н, пос. Целинный, подстанция. В. В. Измалков.

Заочный клуб пользователей БК-0010(01) «Бэкашник» приглашает всех владельцев к сотрудничеству. Напишите нам, и вы получите более подробную информацию. 141008, Московская обл., г. Мытищи-8, а/я 94, ЗКП «Бэкашник». К. А. Виноградов.

У меня есть программы для БПЭВМ «Вектор-06Ц». 393210, Тамбовская обл., г. Рассказово, 3-й Некрасовский пер., 4. Д. А. Мякишев.

Куплю клавиатуру для БК-0010.01. 678730, Я-Саха ССР, Оймяконский р-н, пос. У-Нера, ул. Андрианова, 1, кв. 18. М. Л. Остапчук.

Меняю программы для «Коммодор-64/128» и «Спектрума». Отвечу всем, кто напишет. 143400, Красногорск-4, а/я 54. В. В. Москочев.

Имею около 300 программ для «Спектрума», обмениюсь с другими пользователями. 361000, КБССР, г. Прохладный, ул. Ленина, 85, кв. 24. Р. В. Афанасьев.

Хотели бы переписываться с владельцами БК. Имеем более 500 программ. Готовы поменяться и поделиться. Высылаем каталог. 375069, Ереван, ул. Яна Райниса, 5, кв. 1. В. И. Рогов, А. В. Кузнецов.

Имею множество программ для БК-0010. Программирую на Бейсике и в кодах. Хочу обмениваться программами и переписываться с людьми, интересующимися этим компьютером. 446800, Самарская обл., с. Кошки, квартал 2, д. 4, кв. 18. С. Е. Строкин.

Предлагаю программы к «Спектруму». Для обмена присылайте свой каталог и конверт. Имею 2000 программ. 245110, Сумская обл., г. Шостка, ул. Франко, 24. В. В. Луговой.

Хотел бы переписываться с пользователями пакета MASTER для разработки игровых программ на Бейсике БК-0010.01 и БК-0011. 614051, Пермь, ул. Лякишева, 9, кв. 131. И. П. Березенцев.

Хочу приобрести компьютер, но не знаю, какой выбрать. Прошу владельцев ПК написать мне о своих компьютерах. Отвечу всем. Заранее благодарен. 252187, Киев, ул. Заболотного, 52, кв. 43. Г. Пивторак.

Прошу помочь с программами для «Микроши». 156019, Кострома, Кинешемское ш., 72, школа-интернат № 3. Т. В. Касторнова.

Ищу единомышленников, владельцев ПК «Поиск». Могу предложить электрическую схему компьютера. Имею программы на Бейсике, Паскале, в кодах. Имею также множество программ, книги по ОС и Турбо-Паскалю для «Роботрона-1715». 620072, Екатеринбург, ул. Сыромолотова, 16, кв. 164. А. В. Бутаков.

Имею большую коллекцию программ для бытовых компьютеров «Спектрум» и УКНЦ (МС 0511). Предлагаю обмен программами владельцам «Спектрума» и УКНЦ. Всем заинтересовавшимся вышлю каталоги. 399113, Липецкая обл., Липецкий р-н, с. Частая Дубрава, ул. Московская. С. В. Бирюков.

Куплю игровые программы для ПК «Веста», в. 1.0 (ПК-8000). 659200, Алтайский край, г. Заринск, ул. Воинов-интернационалистов, 3, кв. 58. К. Лоскутов.

Хотел бы обмениваться программами с владельцами БК-0010(01). Имею около 400 программ в кодах. 220074, Минск, ул. Берута, 16, кор. 2, кв. 80. Тел. 52-24-04. А. Славинский.

Хотел бы переписываться с владельцами ПК-86. 288533, Винницкая обл., Чечельникский р-н, с. Любомырка. В. А. Леишук.

Хотел бы обмениваться программами и информацией с владельцами БК-0010.01. Имею программы IBM PC AT/XT. 312050, Харьковская обл., Дергачевский р-н, п/о М. Даниловка, ХЗВИ, д. 11, кв. 61. Ю. С. Саковец.

Кооператив «Электрон» предлагает

владельцам и пользователям ПЭВМ типов «Львов ПК-01», «Вектор-06Ц», «Электроника МС 1502», «Поиск», УКНЦ («Электроника МС 0511»), БК-0010-01, БК-0011, IBM XT/AT, «Специалист», «Синклер ZX Спектрум», «Правец-8Д», ДВК-3/4, РК-86 32К, «Микроша», «Партнер», «Апогей», «Орион», «Агат-7», «Агат-9», «Корвет», «Сура», «Хобби», «Веста», «Ассистент» широкий выбор системных, прикладных, игровых, учебных программ, новейшие разработки из первых рук по умеренным ценам.

Программы, выбранные заказчиком из наших каталогов, высылаются по почте в адрес заказчика. Каталог программ можно получить по почте или купить в кооперативе «Электрон». Для получения каталогов программ и бесплатных рекламных листов необходимо выслать в адрес кооператива «Электрон» открытку с указанием типа компьютера и своего домашнего адреса. В конверт вложите конверт со своим адресом и наклеенными марками. Организациям необходимо выслать гарантийное письмо-заявку в адрес кооператива «Электрон».

Предлагаем большое количество учебных программ для классов УКНЦ («Электроника МС 0202»).

Для ПЭВМ «Львов ПК-01», «Вектор-06Ц», «Поиск», «Электроника МС 1502» имеются также готовые комплекты игровых программ на кассетах. Готовые комплекты можно купить только за наличный расчет и только в помещении кооператива «Электрон».

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

Покупаем программы для ПЭВМ «Львов ПК-01», «Вектор-06Ц», «Апогей», «Поиск», «Электроника МС 1502», «Сура», «Ассистент», «Хобби», «Веста», «Орион»; возможен обмен программами.

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

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

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

Изготавливаем и поставляем:

станок для изготовления стеновых шлако- и керамзитобетонных стеновых блоков. Размер блока 390×200×175 мм, вес станка 750 кг, габариты 1850×1500×1800 мм, напряжение 380 В, производительность 100 шт/ч, одновременное формирование пяти блоков;

деревообрабатывающий станок со следующими возможностями: разрезание продольное, строгание по плоскостям, выборка четверти, изготовление плинтусов, фрезерование пазов. Габариты станка 600×900×700 мм, вес 150 кг.

Высылаем документацию:

комплект рабочей документации для изготовления малогабаритного станка для изготовления шлако- и керамзитобетонных стеновых блоков для желающих изготовить его своими силами;

комплект рабочей документации на деревообрабатывающий станок для желающих изготовить его своими силами;

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

Предлагаем организациям и предприятиям широкий выбор импортной оргтехники для офиса: ксероксы, факсы, телевизоры, видеотехнику, телефоны-автоответчики, микрокалькуляторы, расходные материалы для факсов и ксероксов, дискеты высшего качества американской фирмы 3М.

Для получения бесплатного перечня услуг и товаров с ценами направляйте заявки. Адрес: 103489, Москва, Зеленоград, до остановки «Кинотеатр «Эра», далее автобусами № 1, 2, 6, 7, 10 до остановки «Поликлиника № 65». Вход с торца корпуса 705, кооператив «Электрон». Время работы: с 11 до 18 ч, кроме субботы и воскресенья.

Эксплуатация принтера МС-6312

Принтер МС-6312 при всех своих достоинствах имеет и ряд недостатков. Главный из них — печатающая головка «одноразового» использования. Мы сумели увеличить ее ресурс, произведя небольшую операцию. После использования всех чернил заглушка на передней части головки извлекается с помощью шила. В образовавшемся отверстии нарезается резьба М3, и подбирается соответствующий винт для замены заглушки. Через отверстие в полость головки с помощью шприца вводятся чернила (например, «Радуга-2») любого цвета, винт завинчивается и заливается лаком. Головка снова готова к использованию.

Часто головки текут. Если протечка происходит в месте соединения передней стенки с резервуаром, то щель необходимо промазать клеем или хотя бы пластилином. Хуже, если течет из-под матрицы. В этом случае необходимо осторожно, чтобы не сломать керамическую основу, отделить матрицу от корпуса, очистить гнездо и саму матрицу от остатков клея, обезжирить поверхности и

склеить их по общим правилам клеем «Момент» или «Контактол».

Еще один недостаток принтера — отсутствие рукоятки вращения барабана — так же легко устраняется в домашних условиях.

Вращающий момент от мотора на барабан передается с помощью трех шестеренок, расположенных на принтере справа. Самая большая — средняя; к ней и прикрепляется рукоятка. Для этого достаточно просверлить на шестеренке два — четыре отверстия и привинтить через них рукоятку винтами М2, не забыв проложить кольцевую прокладку.

И еще два маленьких усовершенствования, делающих МС-6312 более удобным: на откидывающейся крышке вдоль отверстия для бумаги нанесите мерную шкалу; на этой же крышке (с внутренней или наружной стороны — по вкусу) приклейте фототаблицу с кодами управления принтером (желающим можем выслать таблицы с кодами для МС-6312 и таблицы восьмеричных и десятичных кодов для БК-0010(01)).

В. РОГОВ, А. КУЗНЕЦОВ

Калейдоскоп БК

А. А. Войлуков (Москва) сообщает о назначении бита 1 ячейки 177712 (см. статью А. Подгорецкого о системном таймере в «ИНФО» 6—91): при его установке или очистке значение из ячейки 177706 копируется в ячейку 177710.

Шебалкин: введение в программу строки

```
10002 POKE &HBC, M%
```

позволяет получать разноцветный шрифт. В качестве М % можно попробовать &HFF55, &HAAFF, &H55FF, &HFFAA.

Полку печатающих устройств БК прибыло: А. Карсалов из г. Лаишево сообщил, что им произведено подключение телеграфного аппарата и написан соответствующий драйвер. Несколько позже такое же сообщение пришло от С. Таланова (Кировская обл., г. Яранск).

Несколько читателей «ИНФО» сообщили, что способ избавления от дребезга клавиату-

ры, предложенный Б. Фролкиным в № 6—90, не дает хороших результатов.

И снова — отключение звукового сигнала, отмечающего нажатие клавиш БК. Программа, предлагаемая О. Барсуковым (г. Умань), отличается малой длиной и дает возможность определить, по какому регистру нажата клавиша.

```
10 DATA &012767, &0101150, &044, &0403,
&012767, &0101374, &034, &04437,
&0110340, &010546, &05737, &0224,
&01402, &04737, &0110536, &012737,
&0177777, &0222, &0113700, &0177662,
&0137, &00
```

```
20 A%=&0414
30 FOR I%=0% TO 21%
40 READ X%
50 POKE A%+I%*2%, X%
60 NEXT I%
70 POKE &060, A%
80 POKE &0274, A%+&010
```

Если клавиша нажата по верхнему регистру, то в ячейке A % + &052 будет число &0101150, если по нижнему, то &0101374.

Продемонстрировать работу программы на примере можно, добавив к ней строки

```
90 ? AT(0,0) OSTR(PEEK(A%+&052))
                                     PEEK(-78%)
100 GOTO 90
```

В. Смирнов из Москвы предлагает три программы. Первая позволяет выводить текст «бегущей строкой».

```
10 CLS
20 Y%=22%
30 AR="ЖЖ Информатика и образование ЖЖ"
40 FOR J%=0% TO 28%
50 ? AT(30%-J%,Y%) MID$(AR,1%,1%+J%)
60 FOR J1%=0% TO 900%
70 NEXT J1%,J%
80 FOR J%=0% TO 28%
90 ? AT(0%,Y%) MID$(AR,1%+J%)
100 FOR J1%=0% TO 900%
110 NEXT J1%,J%
120 GOTO 10
```

62

Текст, который должен кончатся минимум двумя пробелами, будет выводиться в ту строку экрана, номер которой задан в Y %.

Вторая программа позволяет стереть имеющуюся на экране информацию горизонтальной линией, перемещающейся снизу вверх.

```
10 FOR AX=240% TO 0% STEP -1%
20 PSET(0%,A%),1%
30 DRAW"R255"
40 OUT -50%,64%,1%
50 OUT -50%,64%,0%
```

```
60 PSET(0%,A%+1%),0%
70 DRAW"C0R255"
80 OUT -50%,64%,1%
90 OUT -50%,64%,0%
100 NEXT A%
```

Для стирания сверху вниз меняйте A % от 0 до 240 и поменяйте в строке 60 плюс на минус. Для уменьшения частоты звука уберите строки 40 и 50 или 80 и 90, для полного «обеззвучивания» — все четыре перечисленные строки. «Спрятать» стирающую линию можно, сделав строку 30 такой же, как 70.

Третья программа, основанная на идее А. Ланеева (ИНФО. 1991. № 3), выводит на экран слово PROGRAM, плавно меняющее цвет с красного на зеленый и обратно, пока не будет нажата какая-либо клавиша.

```
5 COLOR 1,4
10 CLS
20 ? "PROGRAM"
30 FOR PX=1% TO 8%
40 FOR RX=1% TO 58%
50 IF POINT(RX,PX)=1% THEN
                                     PSET(RX,PX),2%
60 NEXT RX,PX
70 FOR YX=1% TO 8%
80 FOR XX=1% TO 58%
90 IF POINT(XX,YX)=2% THEN
                                     PSET(XX,YX),1%
100 NEXT XX,YX
110 AR=INKEY$
120 IF AR="" THEN 30
130 END
```

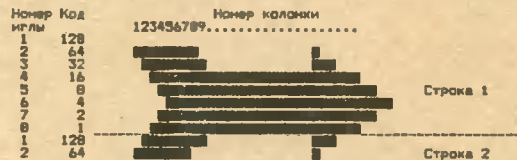
И. ПАНЧЕНКОВ

Печать графической информации на принтерах МС-6312 и МС-6313

Устройства вывода печатающие (УВП) МС-6 12 и МС-6313 позволяют регистрировать на бумаге не только символьную, но и графическую информацию. Для этого они должны быть переведены в графический режим соответствующей управляющей последовательностью — графической командой, после чего принимаемые коды начинают интерпретироваться не как коды символов, а как данные, непосредственно управляющие иглами печатающей головки.

Рисунок, полученный с помощью УВП, состоит из точек (на рис. 1 показано графическое изображение в увеличенном виде) — следов ударов игл печатающей головки МС-6313 или капель чернил из головки МС-6312. За один горизонтальный проход головка печатает строку из 6 (у МС-6313) или 8 (у МС-6312) точек. Перед следующим

проходом бумага сдвигается на 6/72 (у МС 6313) или 8/72 (у МС-6312) дюйма.



Графический режим МС-6312. Этот принтер имеет следующие графические команды: ESC K N1 N2 — одинарная плотность; ESC L N1 N2 — двойная плотность; ESC Y N1 N2 — двойная плотность с высокой скоростью;

ESC *m N1 N2 — универсальная.

Здесь N1, N2 — число колонок; m — число от 0 до 7, задающее режим печати (по-

следней командой можно заменить любую из трех первых). Работа команд достаточно хорошо описана в инструкции по эксплуатации УВП.

Режимы 4—7 — специальные, они изменяют плотность в горизонтальном направлении для обеспечения пропорциональности изображению на дисплее или для согласования с вертикальной плотностью так, чтобы окружности получались круглыми (режим графопостроителя).

Некоторые затруднения у пользователей вызывает резервирование колонок. В одной строке можно отпечатать до 1958 колонок точек, но УВП использует лишь числа, не превышающие 255, так что для резервирования колонок приходится задавать два числа. N2 определяет полное число наборов по 256 колонок, а N1 — остаток от деления нужного числа колонок на 256. Например, для резервирования 200 колонок под графические данные нужно задать N1=200, N2=0; для резервирования 1632 колонок N1=96, N2==6: $96+6 \times 256=1632$.

N1+N2×256 чисел, следующих за графической командой, УВП интерпретирует как графические данные, т. е. управляет разбрызгиванием чернил из сопел печатающей головки в соответствии со значениями битов двоичного представления каждого из этих чисел. Если чисел окажется меньше, чем задано параметрами N1 и N2, УВП остановится и будет ждать продолжения; это создает впечатление блокировки УВП. Если чисел чересчур много, «лишние» будут интерпретироваться как текстовая информация (коды символов).

Приведем пример простейшей программы печати графики.

```
10 LPRINT CHR(27)
20 LPRINT "K"
30 LPRINT CHR(40)
40 LPRINT CHR(8)
50 FOR I=1 TO 40
60 LPRINT CHR(74)
70 NEXT I
```

Строки 10, 20 задают режим одинарной плотности; строки 30, 40 резервируют 40 колонок; строка 60 обуславливает выброс чернил из 2-го, 5-го и 7-го сопел ($74=64+8+2=01001010_2$), что даст в итоге три линии, протянувшиеся на 40 колонок.

Графический режим MC-6313 (версия 081289/MC6307). Этот принтер переводится в графический режим командой Ap2 P n q (в восьмеричном виде 33, 120, n, 161). Здесь p определяет плотность графической печати (от 481 до 1921 колонок в строке).

После этого принтер будет интерпретировать получаемые числа как графические данные, пока не получит команду выхода из графического режима Ap2 \ (в восьмеричном виде 33, 134).

Графические данные интерпретируются следующим образом: из поступившего в УВП числа (оно должно быть от 63 до 126, или, в восьмеричном виде, от 77 до 176) вычитается 63, и в соответствии с битами результата ударяют иголки печатающей головки.

Номер иглы	Код
1	1
2	2
3	4
4	8
5	16
6	32
7	не используется
8	не используется

Например, чтобы в очередной колонке напечатать точки верхней (первой) и третьей иглами, нужно заслать в УВП число 68: $68-63=5=000101_2$.

Несколько советов желающим самостоятельно написать драйвер копии экрана. Он должен быть оформлен в виде подпрограммы, вызываемой по команде JSR PC, и содержать следующие программные модули: определения адреса экранной памяти, соответствующей левому верхнему углу экрана в момент выдачи копии;

задания плотности графической печати; циклического преобразования участка видеопамати в графический образ; передачи кодов на УВП.

Изложенный материал поможет вам самостоятельно писать программы вывода графической информации на принтер, однако гораздо проще использовать готовые драйверы.

А. БАРСУКОВ

Приносим читателям свои извинения за неверную верстку в «ИНФО» 1—92: часть текста, начинающаяся на правой колонке с. 67 словами «Кроме программного счетчика...» и кончающаяся в левой колонке с. 68 словами «...с другими регистрами», должна была быть помещена перед абзацем, начинающимся словами «Отдельные биты...» (с. 69, левая колонка).

На экране — поверхность

Журнал уже обращался к теме изображения трехмерных поверхностей на экране БК (ИНФО. 1990. № 2), однако приведенные в той статье формулы преобразования координат и построенная на этих формулах программа дают только одну проекцию, жестко заданную в самой программе. Естественно, это снижает наглядность.

Здесь описывается программа, позволяющая рассматривать со всех сторон поверхность, построенную с помощью сеточных сечений.

Подобная программа должна содержать следующие блоки:

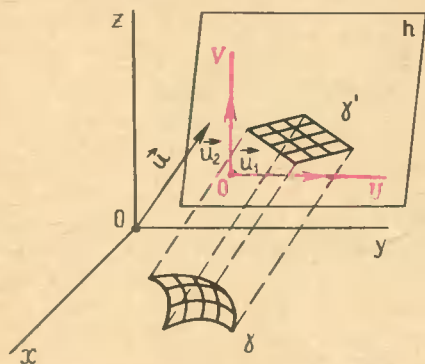
введение направления проекции и перевод координат поверхности из старой системы координат (ССК) в новую (НСК);

блоки, рисующие сечения, параллельные осям OX и OY;

блок, проверяющий, не закрывается ли данная точка другой, расположенной ближе к наблюдателю, т. е. блок контроля видности.

Последний вопрос хорошо разобран в [1] и [3], поэтому здесь мы его рассматривать не будем, а сосредоточимся на построении параллельной проекции (изложение общих принципов см. в [2]).

Пусть в трехмерной системе координат (рис. 1) задана поверхность γ . Направление



проектирования задается единичным вектором $\vec{u} = (a_0, b_0, c_0)$. Плоскость проекции h перпендикулярна вектору u и может находиться где угодно (так как проекция параллельна), а фигура γ' — проекция поверхности γ . Чтобы получить координаты каждой точки фигуры γ' , надо ввести НСК в плоскости проекции OUV и написать формулы преобразования координат

$$U = f(x, y, z);$$

$$V = g(x, y, z).$$

Располагаться НСК в этой плоскости может как угодно; единственное условие — $\vec{u} = \vec{u}_1 \times \vec{u}_2$, где \vec{u}_1 и \vec{u}_2 — единичные базисные векторы НСК. Это условие означает что $\vec{u}_1 \perp \vec{u}$ и $u_2 \perp u$.

Если принять, что плоскость проекции проходит через точку $O(0, 0, 0)$, то формулы преобразования примут вид:

$$U = xa_1 + yb_1 + zc_1 = \vec{r} \cdot \vec{u}_1,$$

$$V = xa_2 + yb_2 + zc_2 = \vec{r} \cdot \vec{u}_2,$$

где \vec{r} — вектор, характеризующий положение текущей точки поверхности в ССК. Координаты векторов \vec{u}_1 и \vec{u}_2 можно вычислить по формулам:

$$\vec{u}_1 = \left(-\frac{b_0}{\sqrt{a_0^2 + b_0^2}}; \frac{a_0}{\sqrt{a_0^2 + b_0^2}}; 0 \right),$$

$$\vec{u}_2 = \left(-\frac{a_0 c_0}{\sqrt{a_0^2 + b_0^2}}; -\frac{b_0 c_0}{\sqrt{a_0^2 + b_0^2}}; \sqrt{a_0^2 + b_0^2} \right),$$

где a_0, b_0, c_0 — координаты единичного вектора направления проекции. Условие единичности является обязательным.

В приводимой программе пользователь сам вводит направление проекции, что позволяет рассмотреть поверхность с разных сторон (рис. 2—4). В качестве примера взят гиперболический параболоид:

$$2z = \frac{x^2}{a^2} - \frac{y^2}{b^2}.$$

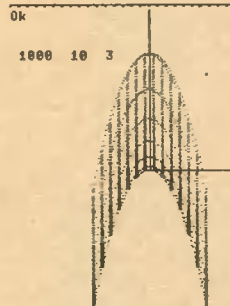
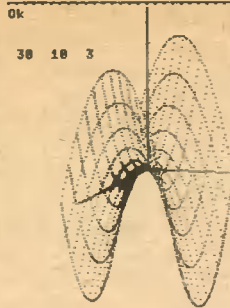
```

5 KEY 1, "BSAVE"+CHR$(34)+"TT:S"+CHR$(34)
6 KEY 2, ".8040004.807776"+CHR$(10)
10 REM "ГИПЕРБОЛИЧЕСКИЙ ПАРАБОЛОИД с комментариями
20 CLS
30 DIM MAX(255), MIN(255)
40 REM "ОПРЕДЕЛЕНИЕ ФУНКЦИИ
50 DEF FNZ(X1, Y1)=X1*X1/40-Y1*Y1/10
60 ? "ВВЕДИТЕ НАПРАВЛЕНИЕ ПРОЕКЦИРОВАНИЯ
70 INPUT A!, B!, C!
80 F1X=SGN(A!+.000001)
90 F2X=SGN(B!+.000001)
100 IF A!=0 AND B!=0 TH A!=-.000001
110 CLS
120 REM "нормирование вектора проекции
125 ? AT(0,3)A!:B!:C!
130 X1=SQR(A!*A!+B!*B!+C!*C!)
140 A1=A1/X1
150 B1=B1/X1
160 C1=C1/X1
170 REM "опр. базисных векторов в плоскости проекции
180 C2!=-SQR(A1*A1+B1*B1)
190 A11=-B1/C2!
200 B11=A1/C2!
210 C11=0
220 A21=-A1*C1/C2!
230 B21=-C1*B1/C2!
240 GOSUB 630
250 GOSUB 680
260 REM "построение сечения, параллельных OY
270 FOR X1=24*F1X TO -24*F1X STEP -4*F1X
280 FOR Y1=-13 TO 13 STEP -1/3
290 Z1=FNZ(X1, Y1)
300 GOSUB 540
310 GOSUB 570
320 NEXT Y1
330 NEXT X1
340 GOSUB 630
350 REM "построение сечения, параллельных OX
360 FOR Y1=13*F2X TO -13*F2X STEP -2*F2X

```

```

370 FOR X1=-24 TO 24 STEP 1/3
380 Z:=FNZ(X1,Y1)
390 GOSUB 540
400 COSUB 570
410 NEXT X1
420 NEXT Y1
425 REM "построение координатных осей
430 LINE (128,120)-(128+50*A11+4.120-50*A21*6)
450 LINE (128,120)-(128+20*B11+4.120-20*B21*6)
460 LINE (128,120)-(128+20*C11+4.120-20*C21*6)
470 BEEP
480 BEEP
490 BEEP
500 BEEP
510 BEEP
515 LOCATE 0,0
520 END
530 REM "расчет координат в плоскости проекции
540 UX=120+(X1*A11+Y1*B11+Z1*C11)*4
550 VX=120-(X1*A21+Y1*B21+Z1*C21)*6
560 RETURN
565 REM "контроль границ экрана
570 IF UX>255 OR VX>240 THEN 620
575 REM "контроль видности
580 IF MIN(UX)<=VX AND VX<=MAX(UX) THEN 620
590 IF VX>MAX(UX) THEN MAX(UX)=VX
600 IF VX<MIN(UX) THEN MIN(UX)=VX
605 REM "построение точки
610 PSET (UX,VX)
620 RETURN
625 REM "инициализация массивов
630 FOR IX=0 TO 255
640 MIN(IX)=240
650 MAX(IX)=0
660 NEXT IX
670 RETURN
675 REM "дополнительный контроль видности
    по краям рисунка
680 Y1=F1X*13
690 FOR X1=-24 TO 24 STEP .25
700 Z1=FNZ(X1,Y1)
710 GOSUB 540
720 MAX(UX)=VX
730 MIN(UX)=VX
740 NEXT X1
750 RETURN
    
```



Параметры и интервалы изменения координат x и y подобраны так, чтобы изображение поверхности не выходило за пределы экрана.

Программа работает следующим образом. В строке 50 определяется функция поверхности — гиперболический параболоид (или любая другая). В строках 60—160 вводится и нормируется вектор направления проекции, а также определяется октант, в котором он находится. Это связано с тем, что для обеспечения контроля видности сечения должны строиться по «уходу» от наблюдателя (см. [3] и [1]), что и реализуется в строках 80, 90, 280, 360. Малые числа, прибавленные к коэффициентам A и B , позволяют избежать нулевых значений координат вектора проекции. Изображения они не меняют в силу своей малости. Строки 180—230 формируют коэффициенты формул преобразования координат (540, 550).

Множители 4 и 6 в 540, 550 играют роль масштаба изображения. Их неодинаковость вызвана тем, что на экране БК при одинаковом

масштабе по вертикали и горизонтали изображение сплющивается по вертикали. Меняя эти коэффициенты, можно изменить размеры изображения.

Можно, изменив строку 50, нарисовать другую поверхность. Можно расширить или сузить область отображения, изменив коэффициенты в строках 270, 280, 360, 370, 680, 690.

Назначение остальных программных блоков понятно из комментариев в тексте программы.

Литература

1. Котов Ю. В. Как рисует машина. М.: Наука, 1988.
2. Фокс А., Претт М. Вычислительная геометрия, применение в проектировании и на производстве. М.: Мир, 1982.
3. Белова Л., Белов Ю. Изображение поверхностей // Информатика и образование. 1990. № 2.

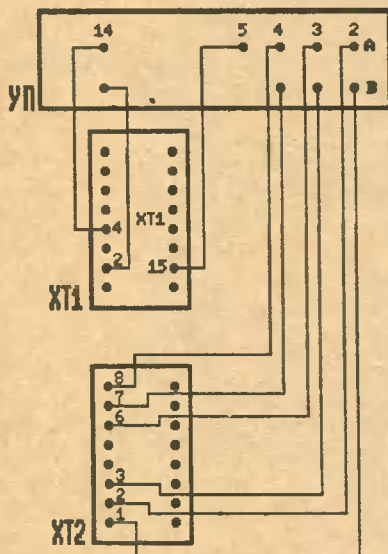
С. ПОГОЖИХ

Джойстик? Нет проблем!

Существует огромное количество игровых программ для БК, предусматривающих подключение джойстика для управления игровыми объектами. Но есть и такие игры, которые не имеют драйвера джойстика. Что же делать, если не хочется портить клавиатуру? Доработать либо программу, либо аппаратную часть.

Первое зачастую гораздо легче, и уже появились доработки известных игр, таких, как New Sheriff или Land. Но иногда вмешательство в программу оказывается невозможным: для драйвера попросту не хватает памяти.

66 Остается второй вариант: подключить джойстик к штатной клавиатуре! Конечно, он потеряет возможность одновременной подачи нескольких не исключаящих друг друга команд (вверх и влево, стрельба и вниз...): штатная клавиатура не допускает одновременного срабатывания нескольких клавиш. Но зато клавиатура будет сохраннее. Кроме того, если изготовить переходник «VII — джойстик», распаяв на один разъем VII два разъема для джойстика, то можно будет подключать один и тот же манипулятор как к порту, так и к клавиатуре. Именно поэтому в описываемой переделке линии клавиатуры выводятся на свободные контакты разъема VII.



На рисунке изображена монтажная схема переделки. Следует учесть, что масштаб и взаимное расположение разъемов не соблюдены, а проводники прокладываются с обратной стороны платы.

Выбранная распайка удобна тем, что соединительные провода не пересекаются, и можно расположить их в промежутке между платой и дном корпуса без каких-либо проблем.

Все соединения выполняются тонким монтажным проводом в пластмассовой изоляции. Пайка должна производиться мало-мощным паяльником, температура жала которого не должна превышать 270 градусов. Само жало необходимо заземлить — микросхемы боятся статического электричества. Особое внимание следует уделить качеству пайки — не должно быть ни малейших потеков припоя! Качественной должна быть и изоляция проводов.

Джойстик распаявается по приводимой таблице, при этом его контакты, соответствующие требуемой клавише, подключаются к указанным выводам разъема VII. Такая распайка позволяет задействовать наиболее употребительные клавиши. На сегодня не существует ни одной игры, заведомо не годящейся для работы с таким джойстиком.

Мы надеемся, что описанная переделка поможет вам дольше сохранить клавиатуру целой. Но это не единственный вариант. Дерзайте!

Всех имеющих какие-либо идеи по этой тематике и просто заинтересовавшихся просим написать по адресу: 241047, Брянск-47, а/я 109. Брянский клуб пользователей БК.

А. МАЛАШЕНКО, Р. АСКЕРОВ

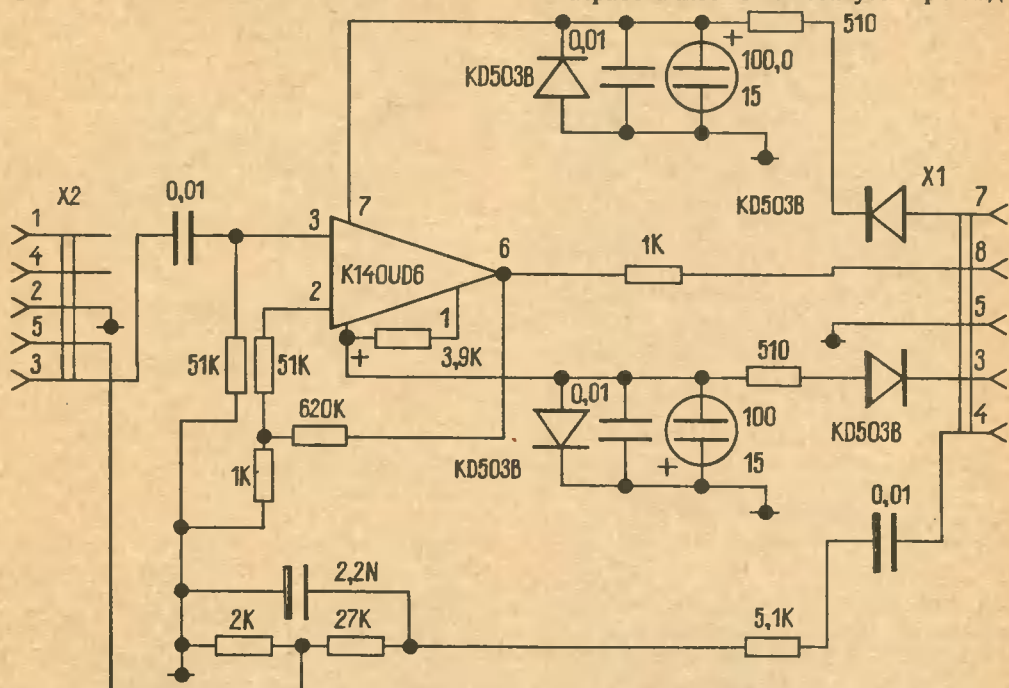
Р. С. За время подготовки статьи выяснилось, что дерзать есть над чем. Завод «Экситон» выпустил БК с новой, более удобной клавиатурой. При этом изменились конфигурация печатной платы и число разъемов, что делает данную переделку невозможной на новых моделях БК. Используя идею, вы можете разработать подключение джойстика и к новой версии.

«Электроника БК-0010» и компьютеры, совместимые с IBM PC

На рисунке приведена схема адаптера, который настолько прост, что его может изготовить даже начинающий радиолюбитель.

читать в ваш компьютер файлы, подготовленные на любых бытовых компьютерах, имеющих выход на магнитофон.

Разработанное в Институте прикладной



Разъем X1 — порт RS-232 (COM1), разъем X2 — обычный магнитофонный разъем с распайкой под «Электронику БК-0010». Сигналы PTS (7-й контакт X1) и TX (3-й контакт) используются для питания операционного усилителя адаптера. Программно на них устанавливается +12 в. (на выходе PTS) и -12 в. (на выходе TX).

С помощью данного адаптера можно подключить к компьютеру, имеющему порт RS-232, любой бытовой кассетный магнитофон. Это дает возможность:

- работать дома на дешевом БК и в дальнейшем переносить наработанную информацию в IBM PC для окончательного редактирования, распечатки, хранения на других носителях и др.;

- использовать все возможности IBM PC как файлового сервера для ваших БК-программ;
- переносить графическую и текстовую информацию, наработанную на IBM PC, в ваш домашний компьютер;

- хранить вашу информацию на компакт-кассете (0,5—2,0 Мб);

- проводить исследования по речевому вводу и выводу;

- математики им. М. В. Келдыша РАН программное обеспечение на IBM PC позволяет: преобразовывать тексты файлов БК-0010 в стандартный plain ASCII;

- дизассемблировать на вашем компьютере программы БК-0010;

- читать с кассеты файлы в стандартном формате БК-0010;

- записывать на кассету файлы в стандартном формате БК-0010.

Набор программ для обслуживания адаптера включает в себя следующие программы:

- | | |
|------|---|
| gm7 | чтение файла с МЛ и запись его на диск с указанием в конце файла исходного имени, адреса загрузки и длины (в формате БК-0010); |
| gmrc | чтение файла с МЛ и запись его на диск как он есть; |
| wrbk | чтение файла с диска и запись его на МЛ в формате БК-0010. Требуется, чтобы в файле присутствовал заголовок в формате БК. Программа предназначена для пересылки на БК через МЛ файлов, первоначально подготовленных на БК и введен- |

ных в РС с помощью программы gm7;

wgrc

чтение с диска любого файла (в том числе созданного программой wgrc) и запись его на МЛ в формате БК-0010. В отличие от программы wrbk не требует никакого специального формата дискового файла. Заголовки для БК-0010 при записи на МЛ воссоздаются по имени файла на РС и может быть скорректированы вручную. Программа предназначена для пересылки на БК файлов, подготовленных на РС, а также для хранения на МЛ файлов РС. Ввод такого файла с МЛ обеспечивается программой wgrc. Обе программы записи на МЛ позволяют выбирать плотность записи;

proct

программа выдает ДАМП файла в восьмеричном виде;

68 bkdisass программа дизассемблирует испол-

няемые БК-файлы и создает файл с расширением .asm. Имеется возможность выбирать различные режимы дизассемблирования для различных участков обрабатываемого файла: в командном режиме файл интерпретируется как последовательность выполняемых команд, а в текстовом — как текстовая константа. Кроме того, дизассемблер можно настроить на различные наборы команд и различные их мнемоники;

bkdocpc

программа преобразует тексты файлов БК-0010 в стандартный plain ASCII на IBM РС. Формат входного файла распознается автоматически, но может быть также задан вручную.

Телефоны для справок и предложений по сотрудничеству: (8-095) 333-61-89, 333-71-89.

В. Сулханов, А. Ходулев.

Московский клуб БК

В октябре 1987 г. группа энтузиастов-пользователей бытовых компьютеров семейства «Электроника БК», считая это семейство компьютеров перспективным как для применения в быту, обучении, так и в научных и исследовательских работах, организовала и юридически оформила Московский клуб пользователей БК.

Главная задача клуба — пропаганда и бескорыстная, активная поддержка бытовых компьютеров семейства «Электроника БК» для использования их в различных сферах деятельности.

В фонде клуба БК насчитывается более 700 разнообразных программ для «Электроники БК» как в машинных кодах, так и на языках высокого уровня. В фонде имеются также методики и рекомендации по усовершенствованию и развитию аппаратной части БК, их сопряжению с телевизорами различных моделей, в том числе цветными. Разработаны несколько вариантов джойстика, различные внешние устройства. Обобщается опыт использования БК в различных целях. Организованы циклы лекций, проводятся тематические консультации.

Большую помощь в становлении клуба оказал Анатолий Семенович Бакланов, пред-

седатель научно-производственного кооператива «Услуга-2». Кооператив оплачивает аренду помещения, помогает создавать необходимые условия для нормальной деятельности Московского клуба БК. Такая помощь дает возможность привлекать в клуб учащихся школ, ПТУ, техникумов и создавать для них творческую атмосферу.

Московский клуб БК поддерживает постоянные связи с аналогичными клубами, группами пользователей и отдельными владельцами «Электроники БК» более, чем в 60 городах. Мы приглашаем и Вас к сотрудничеству и обмену опытом, разработками, программами, чтобы сделать все возможное для сохранения отечественного бытового компьютера «Электроника БК-0010» и других того же семейства. Адрес для переписки: Москва, 129337, а/я 125ж, Московский клуб БК.

В настоящее время в клубе открылось отделение, объединяющее владельцев УКНЦ. Контактные телефоны:

отделение БК 183-18-39

Павел Бенарович

Эльтерман

отделение УКНЦ 333-61-89

Виктор Иванович

Сулханов

Занимательные игры со строковыми функциями

Мне хотелось бы привести несколько программ, которые могут быть полезны при изложении темы «Строковые функции и переменные» на уроках информатики. Некоторые из них можно использовать как эффективные фрагменты-подпрограммы при создании различных программ.

Достаточно часто в различных программах используется «бегущая строка». Задаваемые параметры: X1, Y1 — координаты левого края окна, в котором осуществляется движение строки, D — размер окна.

```
5 CLS: X1=20: Y1=10: D=20
10 INPUT "Введите строку";S$
15 LOCATE,,0: S$=SPACE$(D)+S$+" "
20 N=LEN(S$)
30 FOR I=1 TO N
40 LOCATE X1,Y1: PRINT MID$(S$,I,D)
50 FOR K=1 TO 40: NEXT
60 IF INKEY$<>" " THEN LOCATE,,1: END
70 NEXT I
80 GOTO 30
```

Довольно эффектно программа, реализующая падающие буквы. Из текста, размещенного в верхней строке экрана, с левого края начинают по очереди выпадать буквы и перемещаться в нижнюю строку.

```
10 CLS: INPUT "Введите строку";S$
20 CLS: LOCATE 6,1: PRINT S$
25 N=LEN(S$): LOCATE,,0
30 FOR I=1 TO N
40 FOR M=2 TO 15
45 LOCATE 5+I,M: PRINT MID$(S$,I,1)
50 LOCATE 5+I,M-1: PRINT " "
60 NEXT M
65 FOR K=1 TO 40: NEXT
70 IF INKEY$<>" " THEN LOCATE,,1: END
80 NEXT I
90 GOTO 30
```

Неизвестно, когда дети придумали свой язык, в котором слова произносятся не слева

направо, а наоборот. Программа, приведенная ниже, осуществляет «переворот» слов.

```
10 CLS: INPUT "Введите строку";S$
20 N=LEN(S$)
30 FOR I=0 TO N-1
40 X$=X$+MID$(S$,N-I,1)
45 NEXT I
50 PRINT X$
```

Большой интерес у школьников вызывает программа, прообразом которой послужила игра «Поле чудес». Один из играющих загадывает слово, а второй пытается его восстановить.

```
10 DIM SL$(50),S1$(50)
20 CLS: INPUT "Введите слово
или фразу";S$: CLS
30 N=LEN(S$): LOCATE,,0
40 FOR J=1 TO N
50 SL$(J)=MID$(S$,J,1)
60 NEXT J
70 FOR J=1 TO N
80 IF SL$(J)=" " THEN S1$(J)=" " ELSE
S1$(J)="*"
90 NEXT J
100 GOSUB 300
110 LOCATE 6,5: PRINT "Введите букву"
120 I$=INKEY$: IF I$="" THEN 120
130 FOR J=1 TO N
140 IF I$=SL$(J) THEN S1$(J)=I$
150 NEXT J
160 GOTO 100
300 LOCATE 10,3
310 FOR J=1 TO N
320 PRINT S1$(J);
325 IF SL$(J)<>S1$(J) THEN FL=1
330 NEXT J
340 PRINT
350 IF FL=0 THEN LOCATE,,1: END
360 FL=0
370 RETURN
```

Эта программа может послужить основой для создания компьютерного варианта телевизионной игры «Поле чудес».

Г. КОРНИЛОВ
Санкт-Петербург

Сетевая операционная система для УКНЦ

Одним из несомненных достоинств микро-ЭВМ УКНЦ (МС 0511) является возможность использования ее в составе малой вычислительной сети. Штатные сетевые программные средства, входящие в комплект поставки КУВТ УКНЦ 0202, обеспечивают загрузку с РМП по кольцевому каналу связи на каждое из 12 РМУ автономных SAV-программ, а также чтение и запись из загруженных программ файлов с дисков РМП.

По уровню диск- и файл-сервиса автономные системы типа Бейсик значительно уступают операционной системе (ОС) РМП. Объясняется это тем, что автономные системы, будучи ОЗУ-резидентными, вынуждены полностью укладываться в рамки имеющейся оперативной памяти и могут развивать некоторые свои возможности только в ущерб развитию других.

Одним из способов расширения и улучшения файл-сервисных возможностей РМУ при работе с дисками РМП является загрузка на РМУ ОС, аналогичной ОС ФОДОС РМП. Это позволило бы ученикам пользоваться богатым арсеналом номенклатурных программных средств ОС, а также, что немало важно, передать ученическую ЭВМ «в руки» надежного и экономного хозяина, каким давно зарекомендовала себя ФОДОС (стаж ее работы на машинах совместимой архитектуры приближается к 20 годам). К сожалению, в рамках возможностей штатных программных средств эта задача не может быть решена удовлетворительно.

Как загрузить ФОДОС на РМУ? Какими программными средствами для загрузки ОС по сети должны быть оснащены микро-ЭВМ?

Ответ на этот вопрос известен. Программным средством, организующим разделение в сети общих ресурсов (дисков, принтера, модема), а также взаимодействие микро-ЭВМ при их совместной работе в сетевой обстановке, должна быть сетевая операционная система (СОС). Рассмотрим на примере одну из таких систем.

СОС КОРНЕТ предназначена для создания малых вычислительных сетей на базе микро-ЭВМ УКНЦ «Электроника-85», ДВК и является сетевым расширением применяемой на этих машинах ОС ФОДОС (RT-11). Она

поддерживает работу абонентских систем (АС) трех типов — серверов, рабочих станций (РС) и АС комбинированного типа, совмещающих возможности серверов и РС. АС-серверы, концентрируя на себе периферийные, информационно-вычислительные и другие общие ресурсы сети, могут предоставлять их в разделяемое пользование другим абонентам сети. Для обеспечения доступа из сети к их ресурсам серверы оснащаются соответствующими обслуживающими сетевыми программными средствами. Специальными «клиентскими» средствами должны оснащаться и абоненты РС.

Основная функция СОС КОРНЕТ — дисковый сервис. АС, оснащенная дисковыми ресурсами достаточного объема, скорости и надежности, может предоставлять свои диски в пользование другим АС. Дисковое пространство, доступное всем абонентам, называется сетевыми дисками, а АС, выделившая свои диски в общее пользование, — диск-сервером сети. В качестве сетевых могут быть назначены любые дисковые устройства сервера, в том числе виртуальные. Число сетевых дисков не может превышать восьми.

Обслуживание клиентов на диск-сервере реализуется с помощью двух программ — драйвера сервера и сетевой сервисной программы, функционирующей как основное задание АС. АС, выполняющая функции диск-сервера, параллельно с оперативным обслуживанием дисковых запросов свободна в фоновом разделе для решения своих задач.

Доступ к сетевым дискам с РС осуществляется с помощью устанавливаемого в их ОС драйвера сети с интерфейсом виртуального диска. Сервис виртуального диска позволяет РС рассматривать сетевые диски как расширение своего дискового пространства и пользоваться при работе с их каталогами стандартными утилитами DIR, PIP или Smirnov Command (файл-сервис ФОДОС). Сервис виртуального диска позволяет РС использовать сетевые диски в качестве системных, причем РС, лишенные дисков вообще, могут грузить себе ОС в режиме «холодного» старта, т. е. сразу после включения питания на ПК.

Вторая важная функция СОС — под-

держка сетевого межадачного взаимодействия. В КОРНЕТе оно основано на механизме вызова удаленных процедур, который позволяет использовать вычислительные ресурсы сервера. РС может через (сквозь) сеть вызвать процедуру, которая будет исполняться на сервере. Обслуживающие части распределенных программ (например, принт-сервер) реализуются на сервере в виде виртуальных программных процессоров. Управление процессорами осуществляется из «клиентских» частей программ на РС.

СОС КОРНЕТ позволяет организовать сети звездно-кольцевой топологии из 250 АС. Связь АС в кольцевой коммуникационной подсети осуществляется с использованием сетевых адаптеров СА-2, в звездообразной — при помощи устройств радиальной связи ИРПС.

Базовая версия СОС поддерживает конфигурацию КУВТ УКНЦ 0202 на 16 АС, обеспечивая загрузку по сети на РМУ ОС ФОДОС с дисков РМП и работу на РМУ в среде ОС. Сетевые операции на РМУ могут выполняться асинхронно, не нарушая одновременной работы с сетью других. Номер сетевого диска, с которого ученик хочет загрузить ОС, задается независимо на каждом РМУ двумя микропереключателями. При наличии на диске сетевой модификации монитора ФОДОС (модификация производится при установе СОС) он может служить системным одновременно для 16 РМУ.

Для повышения надежности передачи и обеспечения равноправного доступа к передающей среде в СОС применяются протоколы, гарантирующие обслуживание РМУ на

уровне пакета — единицы обмена размером не более 128 байт. Для увеличения пропускной способности сети в СОС со стороны диск-сервера используется многопользовательский режим обслуживания, позволяющий одновременно обрабатывать до 8 дисковых запросов, причем каждой транзакции (запрос в стадии обработки) для обмена с диском выделяется буфер размером всего в один дисковый блок.

Поддержка звездообразной сети на базе сервера-ДВК, оснащенного многоканальным устройством радиальной связи, реализована по той же схеме. Сетевой сервис для обеих сетей унифицирован.

Развитие возможностей СОС КОРНЕТ может предусматривать:

реализацию поддержки АС комбинированного типа, что позволило бы, например, назначать в качестве сетевого принт-сервера одну из РС, разгружая от этих функций главный сервер сети;

разработку набора системных сетевых программ, в том числе многопользовательских СУБ, которые позволят применять сеть УКНЦ при построении маломощных распределенных информационно-вычислительных систем, автоматизированных систем управления, систем экспериментальных исследований;

обеспечение доступа к ресурсам серверов на базе ЭВМ с другим типом ОС, например MS-DOS, RSX-11, VAX/VMS.

В настоящее время СОС может использоваться в учебных заведениях при организации лабораторных практикумов по курсу «Локальные вычислительные сети».

А. ГУСЕВ

Операционная система ФОДОС

8.1. Текстовый редактор WRITER

Текстовый редактор WRITER предназначен для подготовки текстовых документов (отчетов, статей и т. д.). Встроенная подсказка облегчает работу с ним и позволяет использовать на уроках информатики. Рассылать редактор следует с помощью сетевой системы SOSNET. Для этого надо:

1. Загрузить систему на РМУ.
2. Назначить рабочим диском устройство VD2, что будет соответствовать диску MZ1:.
3. Поставить в нижний карман дисководов диск с текстовыми файлами.

4. Набрать RUN VD1:WRITER и нажать на клавишу <ВВОД>.

Далее приводятся команды, набираемые на РМУ:

```
VD0>SET USR SWAP
VD0>ASS VD2: DK:
VD2>RUN VD1:WRITER
```

После появления картинки нажать на клавишу <СБРОС>. На экран выводится корневое меню. Клавишами управления курсора выбрать нужный режим работы и нажать на <ВВОД>. Нажатие на клавишу <СБРОС> возвращает в корневое меню. Во всех режимах работы можно воспользоваться подсказкой, для чего дать УПР/П (нажать на кла-

Продолжение. Начало см.: Информатика и образование. 1992. № 2, 3—4.

вишу <УПР> и, не отпуская ее, на клавишу <П>).

В режиме работы с диском текст можно считывать с диска (для его редактирования) и записывать (набранный в режиме РЕДАКТИРОВАНИЕ). Редактор позволяет выводить на экран все текстовые файлы, хранящиеся на диске. Для этого надо перейти в режим РАБОТА С ДИСКОМ, выбрать строку ЧТЕНИЕ ФАЙЛА, нажать на клавишу <ВВОД>, на запрос имени файла ввести символ (?), и нажать на <ВВОД>. На экран выводятся имена файлов. Клавишами управления курсора следует выбрать нужный файл и нажать на <ВВОД>. Для редактирования загруженного файла следует выйти из данного меню и перейти в режим редактирования.

Редактор WRITER позволяет обучать учащихся, но использовать его для создания документов неудобно, так как он создает выходной файл в нестандартном для ОС виде. В этом случае лучше использовать текстовый редактор EDIC.

8.2. Текстовый редактор EDIC

Текстовый экранный редактор EDIC является наиболее мощным текстовым редактором для машин серии УКНЦ и ДВК. Он прост в обращении, но в то же время позволяет использовать все возможности ОС. С его помощью можно готовить документы для печати, используя управляющие символы печатающего устройства. Редактор находится на системном диске. Для запуска набрать EDIC и нажать на клавишу <ВВОД>. Редактор запускается нажатием на функциональную клавишу <K4>. После запуска выдается приглашение:

>>

Требуется нажать имя редактируемого файла. Расширение имени файла можно не указывать. В этом случае редактор будет искать файл с указанным именем и расширением LST. Если на диске нет файла с данным именем, то EDIC выдаст запрос на создание файла:

```
?NYS-I-File not found, Create it ?
```

Следует нажать на клавишу <Y>, а затем на <ВВОД>. Далее приводится пример команд для редактирования файла TEST. LST:

```
MZ0>EDIC
```

(можно заменить нажатием на клавишу <K4>)

```
>>TEST
```

После имени файла можно указать ключи. Полное описание команд и ключей редактора находится на диске с документацией в файле EDIC.DOC. Далее будет рассмотрен

ключ 1, позволяющий просматривать файлы без изменения их содержимого. Таким образом можно просматривать документацию, не распечатывая ее на принтере. Например, для просмотра документации по текстовому редактору EDIC следует поставить в MZ1: диск, содержащий файлы документации, и ввести следующие команды:

```
MZ0>EDIC  
>>MZ1:EDIC.DOC/I
```

Для окончания редактирования следует нажать последовательно клавиши <ИСП> и <E>. При этом редактор выдаст запрос на выход: Exit?. Надо нажать на клавишу <Y>, а затем на <ВВОД>. Отредактированный текст будет записан на диск.

Предусмотрен аварийный выход, без изменения содержимого текста. Для этого надо последовательно нажать на клавиши <ИСП> и <Q>, а затем на запрос редактора

```
Quit, are you sure ?
```

ответить <Y>.

Если вызываемый файл не помещается полностью в ОЗУ, то редактор выдаст сообщение:

```
EOF not in text buffer
```

в ответ следует нажать на клавишу <ВВОД>.

Загруженная часть файла называется страницей. Для подгрузки следующей необходимо дойти до конца страницы и последовательно нажать на клавиши <ИСП> и <N>. Редактор выдаст сообщение:

```
Next ?
```

Следует нажать на <Y>, а затем на <ВВОД>.

При использовании редактора EDIC на РМУ рекомендуется создавать файл на электронном диске, а затем копировать его на МД.

8.3. Программа печати документов D100

Программа печати документов D100 позволяет переводить текст из кода КОИ-7 в КОИ-8, изменять параметры печати, выводить документ на печать. Самый первый документ, который следует распечатать, — README.HLP. В нем находится необходимая информация о работе с данным пакетом программ. Перед печатью не забудьте включить принтер и, если надо, запустить программу D100M (см. п. 8.4.1). Программа печати D100 находится на системном диске. Для ее запуска следует набрать D100 и нажать на <ВВОД>:

```
MZ0>D100
```

На экране появится корневое меню и запрос номера режима работы:

7D100-I—Укажите режим работы:

- 1 — Преобразование текста из КОИ-7 в КОИ-8.
- 2 — Вывод документов на печать.

7D100-Q—Режим:

Для перевода из КОИ-7 в КОИ-8 наберите 1 и нажмите на <ВВОД>. Программа выдаст запрос на ввод имени входного файла:

7D100-Q—Выходной файл:

Следует набрать устройство, где находятся файл и имя файла с расширением, например: MZ1:README.HLP. Затем появится запрос на ввод имени выходного (преобразованного) файла:

7D100-Q—Входной файл:

Следует набрать устройство, где будет находиться преобразованный файл, и его имя с расширением, например: MZ1:README.PRN. В конце работы на экран будет выведено сообщение:

7D100-Q—Выход (E). Продолжить (N).

Повторить (R). Возврат в меню (M):

Преобразование текста может потребоваться, если вы набираете программы на Паскале или Ассемблере в редакторе WRITER. Стандартные компиляторы не понимают этот текст, а программа D100 позволяет подготовить его к трансляции.

Для вывода текста на печать следует выбрать режим 2 (режим печати). На экране появится запрос на имя входного файла. Введите его, как и в предыдущем примере. Самый первый файл, который следует распечатать, — это README.HLP. После открытия файла на экране появятся меню режима печати и рабочая информация:

7D100-I—XXXXXXXXXXXXXXXXXXXXXXXXXXXX

- Y — Количество строк в листе < 0 >
- N — Нумерация <ЕСТЬ>
- P — Пауза после листа <ЕСТЬ>
- L — Левая позиция < 0 >
- I — Номер первого листа < 1 >
- S — FF после листа <ЕСТЬ>
- T — LP: /TT:
- H — Помощь
- E — Выход

7D100-Q—LP: >

Программа запрашивает номер листа, с которого следует распечатать. Количество строк в листе — 0 — означает, что конец листа определен в тексте кодом Ctrl/L (для получения более подробной информации см. D100.DOC).

Файл README.HLP разделен на листы в процессе редактирования, кроме того, они уже пронумерованы, поэтому нумерацию надо отключить, для чего набрать <N> и нажать на <ВВОД>. Вставить лист в принтер и дать <ВВОД>. После окончания печати текущего листа на экране появится запрос на продолжение печати:

7D100-Q—LP: >

Поставьте новый лист и нажмите на <ВВОД>. На печать будет выведена следующая страница текста.

8.4. Универсальный драйвер печати

Универсальный драйвер печати позволяет пользователю работать с любым принтером на УКНЦ.

8.4.1. Различные виды принтеров и работа с ними

С классом УКНЦ поставляются различные типы принтеров, имеющих различный интерфейс и разную кодировку символов. Это вызывает большие трудности при работе с печатью. Данный драйвер сам производит перекодировку символов в зависимости от типа принтера. Для настройки интерфейса следует использовать программу D100M. Далее приводится таблица соответствия кодировки символов и интерфейса.

Тип принтера	Кодировка	Использование программы D100M
Robotron D100	КОИ-7	Не требует
CM-6337	КОИ-8	Не требует
D100M	КОИ-8	Требует
Epson	Альтернативная	Требует

Кроме того, принтер CM-6337 может работать в режиме альтернативной кодировки, что позволяет выводить на него псевдографику.

8.4.2. Настройка драйвера на соответствующий принтер

Для настройки драйвера на ваш принтер выполните команду:

R LP.SYS

После этого на экране появится меню:

Введите номер принтера и нажмите на <ВВОД>

- 1 — Robotron
- 2 — D100
- 3 — D100M, CM-6337
- 4 — Epson
- 5 — CM-6337 в режиме дополнительной кодовой таблицы. В этом режиме возможен вывод псевдографики.
- 0 — Выход без изменений.

Номер?

Введите номер, соответствующий вашему принтеру, и нажмите на <ВВОД>. Драйвер автоматически настроится на соответствующий тип принтера. Если у вас принтер CM-6337, то лучше использовать его в режи-

ме № 5 (дополнительной кодовой таблицы). Это даст возможность выводить символы псевдографики и тексты, набранные в редакторе WRITER.

8.4.3. SET-параметры драйвера

SET LP WIDTH=<десятичное число>

число печатаемых символов в строке. Пример:

SET LP WIDTH=75

число символов в строке — 75.

Если печатаемая строка имеет большее число символов, то лишние будут автоматически перенесены в новую строку.

SET LP ASK

постраничная печать текста.

Драйвер будет отсчитывать заданное число строк, останавливать печать и выводить сообщение: «Дальше?» Для продолжения печати следует ввести ⟨Y⟩ и нажать на ⟨ВВОД⟩ (можно просто нажать на ⟨ВВОД⟩), а для отмены печати надо набрать ⟨N⟩ и нажать на ⟨ВВОД⟩.

SET LP NOASK

отмена постраничной печати.

SET LP PAGE=<десятичное число>

установка количества строк в листе для постраничной печати. Драйвер позволяет отмечать конец листа специальным символом Ctrl/L (код 14 в восьмеричной системе счисления), для этого надо указать число строк, равное 0.

Примеры:

SET LP PAGE=55

число строк в листе — 55, и если установлен режим постраничной печати SET LP ASK, то после вывода 55 строк будет выдан запрос на продолжение. (В режиме SET LP NOASK данная команда игнорируется.)

SET LP PAGE=0

конец листа определять по коду Ctrl/L. (В режиме отмены постраничной печати SET LP NOASK данная команда игнорируется.)

SET LP AUTO

автоматический прогон (выброс) листа после окончания печати (работает только в режиме SET LP ASK).

SET LP NOAUTO

отмена автоматического прогона. После окончания печати листа бумага останется на месте.

SET LP FF

разрешить передачу управляющего кода Ctrl/L (прогон листа) на принтер.

SET LP NOFF

запретить передачу управляющего кода Ctrl/L (прогон листа) на принтер. Следует отметить, что если драйвер работает в режиме постраничной печати, то эту команду он выполняет автоматически. В данном случае прогон листа определяется параметром SET LP AUTO.

SET LP LF

разрешить передачу кода «перевода строки» на принтер.

SET LP NOLF

запретить передачу кода «перевода строки» на принтер. При печати текста из редактора WRITER приходится устанавливать на принтере автоматический перевод строки по возврату каретки. Чтобы иметь возможность печати текста из операционной системы, следует запретить передачу кода «перевода строки».

SET LP EDIWR1

режим совместимости редактора WRITER и стандартных редакторов операционной системы. Используя этот режим, можно средствами операционной системы распечатывать тексты, набранные в редакторе WRITER.

SET LP NOEDIWR1

запретить режим совместимости.

SET LP KOI8

установить режим кодировки КОИ-8.

Данная команда выполняется автоматически при настройке драйвера на соответствующий тип принтера.

SET LP NOKOI8

переход в режим КОИ-7.

При настройке на принтер Robotron данная команда выполняется автоматически.

SET LP KOD

установить перевод кода 1 в код 16 и кода 2 в код 17 (числа в восьмеричной системе счисления).

Используется для управления режимом расширенной и узкой печати для принтера D100. Если в тексте стоит код 1, то на принтер будет подан код 16 (расширенная печать), код 2 будет переведен в код 17 (плотная печать). При настройке на принтер D100 данная команда выполняется автоматически.

SET LP NOKOD

отмена перевода.

Данная команда выполняется автоматически при настройке на любой принтер, отличный от D100.

SET LP CSR=<адрес регистра состояния печати>

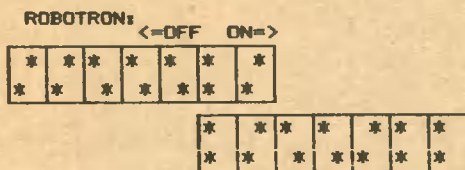
установка адреса регистра состояния печати (для системных программистов).

SET LP VESTOR=<адрес вектора прерывания>

установка вектора прерывания от устройства печати (для системных программистов).

Внимание! Перед использованием принтеров D100M, SM-6337 и Epson следует запустить на выполнение программу D100M. Данная программа настраивает УКНЦ для работы с этими печатающими устройствами.

8.4.4. Установка микропереключателей



8.4.5. Установка автоматического запуска программы D100M при загрузке операционной системы

Установку автозапуска программы D100M следует производить только один раз — либо при загрузке операционной системы при установке даты, либо специально запустить программу

После появления приглашения установки даты:

DATE [29-JUL-91] суббота

```
=====
SET USR NOSWAP
SET EXIT NOSWAP
SET BL ON
RULON
KEY
! < UNUSED >
! < UNUSED >
! < UNUSED >
! < UNUSED >
```

```
=====
-, ,L,H --> команды установки даты
S      --> вход в сервисный режим
E      --> выход
N=Command, N=[1..9]
```

следует войти в сервисный режим. Для этого нажать на клавишу <S>. Курсор перейдет в низ экрана. Теперь можно вводить команды, которые будут выполняться после завершения работы программы DAY. Список уже имеющихся команд выведен на экран под информационной строкой. Вам следует внести в этот список команду запуска программы D100M, для чего определить номер первой свободной строки. В этой строке находится запись:

! < UNUSED >

В нашем примере это строка с номером 6. Теперь следует установить команду. Для этого набрать номер строки, а через знак <=> команду и нажать на <ВВОД>:

*6=R D100M

Введенная строка установится под строкой KEY. Теперь надо выйти из сервисного режима. Для этого следует набрать <E> и нажать на <ВВОД>:

*E

Программа перейдет в режим установки даты. Нажмите на <ВВОД>, и все команды будут выполнены. Программа DAY запишет все установки на диск и при следующем обращении выполнит их.

Для удаления строки из меню команды DAY следует перейти в сервисный режим, набрать номер строки, символ <=> и нажать на <ВВОД>. Например, для удаления строки с номером 3 следует ввести команду:

*3=

Внимание! Если у вас принтер Robotron, то программу D100M запускать (и тем более устанавливать в программу DAY) не надо.

Продолжение следует.

Кроссворд для УКНЦ

Одной из форм активизации познавательной деятельности учащихся является учебный кроссворд. Эта форма может быть реализована в дисплейных классах на уроках информатики и при изучении других предметов.

Рассмотрим один из принципов создания учебных кроссвордов для дисплейного класса УКНЦ.

При составлении кроссвордов следует учесть, что на рабочих местах учащихся на экран нельзя одновременно выводить графику и символьную информацию. Поэтому для изображения на экране сетки кроссворда используются специальные символы, перечисленные здесь с указанием их кодов:

пробел	32		165	+	176
○	36	L	166	—	181
┌	161	┐	170		183
└	163	┘	171	┘	185
	164	└	174	=	186

Вывод этих символов на экран производится оператором ?CHR○ (код).

Экран ЭВМ в дисплейном классе УКНЦ имеет размеры 24 символа по вертикали и до 80 символов по горизонтали (длину строки в 80 символов устанавливают оператором VIDTH 80). Это позволяет разместить на экране кроссворд размером 11 клеток по вертикали и 30 клеток по горизонтали, оставив справа от него чистое поле для комментариев. Такие размеры кроссворда обусловлены тем, что для вывода линий, разграничивающих клетки кроссворда выделяются отдельные строки (столбцы). 11 клеток по вертикали занимают на экране 23 строки (11 букв и 12 линий).

При составлении программы удобно сетку кроссворда очертить на бумаге в клетку и «зашифровать» ее с помощью приведенных выше кодов, учитывая при этом все имеющиеся пробелы. Полученные коды можно разместить в операторах DATA для удобства ставя в соответствие каждому оператору одну строку сетки кроссворда. Вывод на экран реализуется в этом случае операторами READ и ?CHR○ (код). Справа от сетки выводятся тексты, содержащие вопросы к загаданным в кроссворде словам. Анализ набираемого учеником слова удобно сделать в виде подпрограммы. Она должна выделить клетку-начало угадываемого слова (например, обвести второй чертой) и определить реакцию ЭВМ на набранный учеником ответ. При этом предусматривается следующее.

Если ученик, не набирая ответ, нажимает «ВВОД», программа переходит к следующему вопросу. Такой ответ ученика не считается ошибкой, и в дальнейшем у него будет возможность снова вернуться к этому слову.

При неправильном ответе происходит переход к следующему вопросу, в сетке кроссворда изменений нет, а ученику засчитывается одна ошибка.

При правильном ответе угаданное слово вносится в сетку кроссворда, а ученику начисляется одно очко.

При нажатии клавиш «К» (конец) и «ВВОД», указывается количество угаданных слов и допущенных ошибок. Программа завершается, если угаданы все слова кроссворда.

Тексты-комментарии к словам выводятся на экран поочередно, причем после последнего слова программа вновь возвращается к первому и идет снова по кругу, исключая уже угаданные слова.



В качестве примера приведу программу кроссворда по теме «История развития средств вычислений» (рис. 1).

```

10 DIM T(10)
20 FOR I=1 TO 10
30 T(I)=0
40 NEXT I
50 REM SP — кол-во верных ответов, PP — кол-во
  ошибок
60 PP=0
70 SP=0
80 REM рисунок сетки кроссворда
90 CLS
100 DATA 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,
        32, 32, 32, 32, 32, 32, 170, 181, 163,
        32...
. . . . .
310 FOR I=0 TO 20
320 FOR J=0 TO 22
330 READ Z
340 ?AT(J, I) CHR$(Z)
350 NEXT J
360 NEXT I
370 ?CHR$(14)
380 REM слова, загаданные в кроссворде
390 ?AT(30, 2)» по горизонтали :
400 N=1
410 IF T(1)=1 THEN 470
420 X=7
430 Y=3
440 ?AT(32, 4) «французский ученый, сконстру-
  AT(32, 5)«ировавший в XVII веке суммиру-
  AT(32, 6) «ющую счетную машину.
450 F1$=«ПАСКАЛЬ»
460 GOSUB 1170
470 N=2
. . . . .
760 ?AT(30,2)«по вертикали :
770 N=6
780 IF T(6)= 1 THEN 840
790 X=3
800 Y=3
810 ?AT(32,4)«ученый, создатель первой ЭВМ»
  AT(32,5)« в СССР.
820 F1$=«ЛЕБЕДЕВ»
830 GOSUB 1170
. . . . .
1120 GOTO 390
1130 CLS
1140 ?AT(5,10) «правильных ответов — «SP»,
  ошибок — «PP»
1150 ?CHR$(15)
1160 END
1170 REM подпрограмма анализа ответа
1180 ? AT(X,Y-1)CHR$(186)AT(X-1,Y)
  CHR$(164)AT(X+1,Y)CHR$(164)AT
  (X,Y+1)CHR$(186)
1190 ? AT(42,10)
1200 F$=« »
1210 INPUT F$
1220 IF F$=« »THEN 1350
1230 IF F$=«K» THEN RETURN 1130
1240 IF F$=F1$ THEN 1280

```

```

1250 REM неправильный ответ
1260 PP=PP+1
1270 GOTO 1350
1280 REM внесение правильного ответа в сетку
  кроссворда
1290 FOR I=1 TO LEN (F$)
1300 IF N<6 THEN ? AT(X+(I-1)*2,Y)
  MID$(F$,I,1) ELSE ?AT(X,Y+(I-1)*2)
  MID$(F$,I,1)
1310 NEXT I
1320 SP=SP+1
1330 IF SP=10 THEN RETURN 1130
1340 T(N)=1
1350 REM подготовка к следующему слову
1360 ?AT(X,Y)1)CHR$(181)AT(X-1)Y)
  CHR$(183)AT(X+1,Y)CHR$(183)AT
  (X,Y+1)CHR$(181)
1370 FOR I=4 TO 10
1380 ? AT(30,I)spc(33)
1390 NEXT I
1400 RETURN

```

Рассмотрим подробнее строки программы. Строки 10—70 устанавливают начальные значения параметров, используемых в программе. Здесь T (i) — признак, угадано слово (T(i)=1 или нет 'T(i)=0).

Строки 80—370 — вычерчивание сетки кроссворда.

Строки 380—1120 — перечисление слов, загаданных в кроссворде. Здесь X и Y — координаты клетки-начала угадываемого слова, F1\$ — правильный ответ, N — порядковый номер слова.

Строки 1130—1160 — вывод результатов работы и конец программы.

С 1170-й строки начинается подпрограмма анализа ответа. После выделения клетки-начала угадываемого слова (строка 1180) ученик может вводить ответ, и затем ЭВМ выбирает один из перечисленных выше четырех вариантов выполнения программы. Строки 1250—1270 соответствуют неправильному ответу, строки 1280—1340 — правильному. В строках 1290—1310 правильный ответ вносится в сетку кроссворда по горизонтали (при N<6) или по вертикали (при N>=6). В строке 1330 проверяется, разгадан ли весь кроссворд или нет. Строки 1350—1390 осуществляют подготовку для перехода к следующему слову: «гасят» выделенную клетку и стирают текст справа от сетки кроссворда.

Предлагаемая программа достаточно универсальна. На ее основе путем изменения сетки кроссворда (строки 100—300) и комментариев к словам можно получить учебные кроссворды по самым разным предметам школьного курса.

ТЕХНОКОМ

предлагает

НОВЫЕ РАЗРАБОТКИ ДЛЯ УЧЕБНОГО КОМПЬЮТЕРА УКНЦ

ТЕХНИЧЕСКИЕ СРЕДСТВА

- СТЕНД ДИАГНОСТИЧЕСКИЙ ДЛЯ РЕМОНТА КОМПЬЮТЕРОВ УКНЦ И БК 0010;
- ЭЛЕКТРОННЫЙ ДИСК ЕМКОСТЬЮ 1Мбайт;
- ЭЛЕКТРОННЫЙ ДИСК ЕМКОСТЬЮ 0,5 - 1Мбайт В КОРПУСЕ КМД;
- "ВИНЧЕСТЕР" ЕМКОСТЬЮ 20Мбайт, СОВМЕЩЕННЫЙ С КОНТРОЛЛЕРОМ И НАКОПИТЕЛЕМ НА ГИБКИХ И ЖЕСТКИХ ДИСКАХ;
- BASIC НА КАССТЕТЕ ПЗУ, ПОЗВОЛЯЮЩИЙ ЭКСПЛУАТИРОВАТЬ УЧЕБНЫЙ КЛАСС ДАЖЕ ПРИ ВЫХОДЕ ИЗ СТРОЯ НАКОПИТЕЛЯ И ЛОКАЛЬНОЙ СЕТИ;
- РЕДАКТОР ТЕКСТА НА КАССТЕТЕ ПЗУ С ВЫВОДОМ НА ПЕЧАТЬ, ПОЗВОЛЯЮЩИЙ БЕЗ НАКОПИТЕЛЯ ОПЕРАТИВНО НАБРАТЬ И РАЗМНОЖИТЬ ТЕКСТОВУЮ ДОКУМЕНТАЦИЮ;
- КАССТЕТА ПЗУ С НАБОРОМ ТИПОВЫХ БУХГАЛТЕРСКИХ БЛАНКОВ С ВОЗМОЖНОСТЬЮ ЗАПОЛНЕНИЯ И ВЫВОДА НА ПЕЧАТЬ;
- МОДУЛЯТОР, КОТОРЫЙ ПОМОЖЕТ ВАМ ПОДКЛЮЧИТЬ ИГРОВОЙ КОМПЬЮТЕР ТИПА "VINCLAIR" ИЛИ УКНЦ К ЦВЕТНОМУ ТЕЛЕВИЗОРУ;
- ПРОГРАММАТОР ДЛЯ МИКРОСХЕМ 1801PP1, КОТОРЫЕ ИСПОЛЬЗУЮТСЯ В КАССТЕТАХ ПЗУ УКНЦ.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ УКНЦ В ШИРОКОМ АССОРТИМЕНТЕ !

УНИВЕРСАЛЬНЫЕ КАЧЕСТВА НАШЕГО КОМПЬЮТЕРА ПОЗВОЛИЛИ РАЗРАБОТАТЬ И СЕРИЙНО ВЫПУСКАТЬ ТЕЛЕГРАФНЫЙ АППАРАТ НА БАЗЕ УКНЦ. ОН МОЖЕТ БЫТЬ УСТАНОВЛЕН В ЛЮБОМ УЗЛЕ СВЯЗИ И ЕГО ЭКСПЛУАТАЦИЮ ВПОЛНЕ МОЖЕТ ОБЕСПЕЧИТЬ СПЕЦИАЛИСТ, ЗНАКОМЫЙ С РАБОТОЙ УКНЦ.

ВСЯ ПРОДУКЦИЯ ПОСТАВЛЯЕТСЯ С ПОДРОБНЫМ АВТОРСКИМ ОПИСАНИЕМ!

МЫ ПРОВОДИМ РЕМОНТ УКНЦ И ПЕРИФЕРИЙНЫХ УСТРОЙСТВ ДЛЯ ДРУГИХ ТИПОВ КОМПЬЮТЕРОВ

НАШ ОПЫТ ЭКСПЛУАТАЦИИ УЧЕБНЫХ КОМПЬЮТЕРОВ, ВСПОМОГАТЕЛЬНЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПОЗВОЛЯЮТ СУЩЕСТВЕННО ПОВЫСИТЬ ЭФФЕКТИВНОСТЬ РАБОТЫ УЧЕБНЫХ КЛАССОВ И УСТРОЙСТВ НА БАЗЕ УКНЦ.

НАШ АДРЕС: 103489, МОСКВА, А/Я 51
ТЕЛ. 536-50-90 536-60-29



Пользователям ПЭВМ УКНЦ и типа «ZX Spektrum»!

Научно-производственное предприятие «ЕМЕТ»
предлагает

пакет новых универсальных инструментальных программ для создания системы автоматизированного обучения и контроля знаний по любым предметам.

Пакет предназначен для подготовки заданий с выбором ответов или однозначными ответами, генераторов задач, включая генерирование графиков функций, создание рисунков и других иллюстраций, музыкальных сопровождений с последующим использованием созданных файлов в исполнительных программах.

Пакет содержит редакторы текстов, рисунков, графиков функций, музыкальных произведений, психолого-педагогических тестов и исполнительных программ.

Пакет позволяет подготовить систему автономных файлов для ваших прикладных инструментальных средств и систем, открывает новые резервы и стимулы для собственной инициативы.

Имеются готовые файлы по информатике, математике, неорганической и органической химии, географии (VI—VIII классы), истории древнего мира и средних веков, культуре и искусству, большое количество различных тестов, а также обширный пакет по физике и пакет программ для чувашских школ на родном языке, подготовленные группой творчески работающих преподавателей республики при ИУУ.

По запросу высылается наложенным платежом рекламный вариант на указанном вами магнитном носителе. Стоимость его без учета носителя и пересылки — 50 рублей.

Кроме этого, для ПЭВМ типа «ZX Spektrum» подготовлены русифицированные варианты базы данных (MF 09), графического (ART STUDIO) и музыкального (MUZIC BOX) редакторов.

Каталог высылается бесплатно. Для ускорения обработки корреспонденции просьба вложить конверт с заполненным обратным адресом.

Адрес: 429900, Чувашия, г. Цивильск, ул. Ленина, 6, МНПП «ЕМЕТ».

Петербургский производственный кооператив

«Компьютерные игры»

предлагает

Программное обеспечение для пользователей компьютеров

БК-0010, 0010.01, БК-0011М, ДВК-2М, 3М, УКНЦ (МС-0511), ZX-Spectrum, «Хоббит» ИК8030 (128К), «Поиск», МК-88, IBM PC, классов информатики КУВТ-86, КУВТ-УКНЦ.

Кооператив располагает обширным банком игровых, учебных, системных, прикладных программ.

Расценки вдвое-втрое ниже рыночных.

Списки программ высылаем *бесплатно* всем желающим, но прежде всего тем, кто:

обозначит тип своего компьютера прямо на конверте;
вложит в письмо конверт с наклеенными марками и надписанным обратным адресом.

Для пользователей БК-0010, 0010.01:

турбокопирующий, многократно повышающий плотность и надежность записи, — бесплатно, при условии заказа программ;
недорогой телефонный модем.

Переписка: 189610, С.-Петербург — Ломоносов, а/я 649.

С. АХМАНОВ, А. НЕЧАЕВ, А. СКУРИХИН

Архитектура процессора KP580BM80A

Приступим к рассмотрению арифметических и логических операций. Эта группа команд имеет особое значение, так как именно она позволяет производить обработку данных.

Арифметические операции — это сложение или вычитание аккумулятора и (из) одного из регистров, памяти или числа, заданного в команде, с переносом (заемом) или без, а также увеличение и уменьшение на единицу. Последние операции допустимы над содержанием любого 8-разрядного регистра и ячейки ОЗУ. Кроме того, группа арифметических команд включает команду десятичной коррекции, используемую для преобразования чисел из двоичного формата в двоично-десятичный. Далее будет показано, что этих команд достаточно для обработки любых чисел, в том числе и с плавающей запятой.

Логические операции — это поразрядные операции И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ и сравнение. Операнды, используемые в них, те же, что и в арифметических. Есть также команды, реализующие сдвиг аккумулятора влево и вправо и его побитное инвертирование.

Все арифметические и логические операции воздействуют на регистр флагов, поэтому уместно отнести к этой группе еще две команды — установки и инвертирования флага переноса.

Ниже приведена таблица, представляющая описываемые команды. Помимо той информации, что давалась в командах пересылки, в нее включено воздействие операций на флаги условий.

№ п/п	Мнемоника и длительность выполнения	Код и изменяемые биты регистра флагов	Действие
20	ADI N 7 тактов	1 1 0 0 0 1 1 0 N N N N N N N N Z, S, P, C, AC	Сложение содержимого аккумулятора и 8-разрядного числа
21	ADD R 4 такта	1 0 0 0 0 R R R R Z, S, P, C, AC	Сложение содержимых аккумулятора и одного из регистров общего назначения
22	ADD M 7 тактов	1 0 0 0 0 1 1 0 Z, S, P, C, AC	Сложение содержимых аккумулятора и ячейки памяти, адресуемой через HL
23	ACI N 7 тактов	1 1 0 0 1 1 1 0 N N N N N N N N Z, S, P, C, AC	Сложение содержимого аккумулятора (с учетом флага переноса)

Продолжение. Начало см.: Информатика и образование. 1992. № 1—3.

24	ADC R 4 такта	1 0 0 0 1 R R R Z, S, P, C, AC	с 8-разрядным числом Сложение содержимых аккумулятора (с учетом флага переноса) и регистра общего назначения
25	ADC M 7 тактов	1 0 0 0 1 1 1 0 Z, S, P, C, AC	Сложение содержимых аккумулятора (с учетом флага переноса) и ячейки памяти, адресуемой через HL
26	SUI N 7 тактов	1 1 0 1 0 1 1 0 N N N N N N N N Z, S, P, C, AC	Вычитание из содержимого аккумулятора 8-разрядного числа
27	SUB R 4 такта	1 0 0 1 0 R R R Z, S, P, C, AC	Вычитание из содержимого аккумулятора содержимого регистра общего назначения
28	SUB M 7 тактов	1 0 0 1 0 1 1 0 Z, S, P, C, AC	Вычитание из содержимого аккумулятора содержимого ячейки памяти, адресуемой через HL
29	SBI N 7 тактов	1 1 0 1 1 1 1 0 N N N N N N N N Z, S, P, C, AC	Вычитание (с учетом флага переноса) из содержимого аккумулятора 8-разрядного числа
30	SBB R 4 такта	1 0 0 1 1 R R R Z, S, P, C, AC	Вычитание (с учетом флага переноса) из содержимого аккумулятора содержимого регистра общего назначения
31	SBB M 7 тактов	1 0 0 1 1 1 1 0 Z, S, P, C, AC	Вычитание (с учетом флага переноса) из содержимого аккумулятора содержимого ячейки памяти, адресуемой через HL
32	CPI N 7 тактов	1 1 1 1 1 1 1 0 N N N N N N N N Z, S, P, C, AC	Сравнение содержимого аккумулятора и 8-разрядного числа
33	CMR R 4 такта	1 0 1 1 0 R R R Z, S, P, C, AC	Сравнение содержимых аккумулятора и регистра общего назначения
34	CMR M 7 тактов	1 0 1 1 1 1 1 0 Z, S, P, C, AC	Сравнение содержимых аккумулятора и ячейки памяти, адресуемой через HL

20. Сложение содержимого аккумулятора и 8-разрядного числа:

ADI N

(addition immediate — прямое сложение). В результате этой операции к содержимому аккумулятора прибавляется число N и результат сложения помещается в аккумулятор; кроме того, изменяется (F):

Z — флаг (или бит) нуля — устанавливается в 1, если результат сложения равен 0, и в 0, если результат не равен 0;
S — флаг знака — в 1, если результат отрицательный, и в 0, если положительный;
P — флаг четности — в 1, если в битах результата четное число единиц, и в 0, если оно нечетное;

С — флаг переноса — в 1, если в процессе сложения был перенос из 7-го бита, и в 0, если переноса не было.

Рассмотрим пример. Пусть в аккумуляторе хранилось число 05H. Тогда после выполнения команды

```
ADI 21H
```

будет (A)=26H, (Z)=0, (S)=0, (P)=0, (C)=0.

Действие арифметических и логических команд не столь очевидно, как команд пересылок, поэтому рекомендуем изучать их с помощью специальной программы-отладчика, например, SID или DDT (эти программы входят в состав пакета CP/M).

Отладчик — это инструмент для отладки программ в машинных кодах. Здесь мы не будем подробно описывать функции и приемы его использования (это материал для отдельной статьи), ограничимся лишь необходимым минимумом. Нам нужны будут следующие возможности:

реассемблирование — представление машинных кодов в виде последовательности мнемоник команд процессора. Формат команды:

L — реассемблирование с текущего указателя;

Laddress — реассемблирование с заданного указателя;

LaddressStart, addressEnd — реассемблирование с заданного по заданным указателю;

L, addressEnd — реассемблирование до заданного указателя.

Ассемблирование — исправление кода в мнемонике процессора. Формат команды:

Address — ассемблирование с адреса (address).

После ввода этой команды отладчик ждет ввода команды в мнемонике процессора; получив ее, переводит в код и записывает с указанного адреса. Это он делает до тех пор, пока вы вводите правильные команды или не поставите точку.

Просмотр регистров и изменение содержимого регистров. Формат команды:

X — отладчик выводит содержимое всех регистров на экран;

Xрегистр — отладчик выводит содержимое указанного регистра на экран и ждет новое значение этого регистра. Если вы не желаете изменить содержимое регистра, просто нажмите клавишу ENTER.

Трассировка: пошаговое выполнение программы. Формат команды:

T — выполнится одна команда, находящаяся по адресу, на который указывает счетчик команд.

Отметим, что все числа в отладчике вво-

дятся и выводятся в шестнадцатеричном виде.

Рекомендуем следующую последовательность действий при изучении команд с помощью отладчика. Перво-наперво, естественно, необходимо запустить программу. Затем:

1) с помощью команды ассемблирования ввести изучаемую команду, например:

```
A100  
0100 ADI 21
```

2) установить регистр счетчика команд на адрес нашей команды:

```
XP  
P=XXXX 100 (XXXX — текущее значение)
```

3) установить требуемые значения регистров, используемых в команде:

```
XA  
A=XX 5 (XX — текущее значение)
```

4) убедиться, что все регистры установлены правильно:

X

Отметим, что в напечатанной строке справа отображена мнемоника текущей команды (в нашем примере — ADI 21);

5) выполнить одну команду:

T

6) посмотреть результат:

X

Обратите внимание, как изменилось содержимое регистров и флагов. Попробуйте выполнить одну команду несколько раз с разными значениями используемых регистров и посмотрите, как будет меняться результат.

21. Сложение содержимого аккумулятора с содержимым одного из регистров общего назначения:

```
ADD R код  
код команды  
10000RRR
```

где RRR расшифровываются так же, как в команде MVI R,N (№ 1 в нашем списке).

Содержимое аккумулятора складывается с содержимым указанного регистра общего назначения, и результат помещается в аккумулятор; биты Z, S, P и C регистра F устанавливаются так же, как в предыдущем случае. Например, если в A хранилось число 11H, а в B — 33H, то после выполнения операции ADD B в аккумуляторе будет число 44H, (B) не изменится.

22. Сложение содержимого аккумулятора с содержимым ячейки памяти, адресуемой с помощью HL:

```
ADD M
```

При выполнении этой команды (A) становится равным старому содержимому A, сложенному с (M); биты Z, S, P и C регистра

F устанавливаются так же, как в предыдущем случае.

Пусть (A)=22H, (HL)=1234H, (M)=33H. После выполнения команды ADD M содержимое HL и M (ячейки памяти с адресом 1234H) не изменится, а (A) станет равным 55H.

Мы надеемся, что читатель уже немного привык к терминологии и не нуждается в слишком тщательном разжевывании материала, и будем далее объединять схожие команды для сокращения объема текста.

23—25. Сложение содержимого аккумулятора с учетом флага переноса. Операндами здесь могут быть: 8-разрядное число,

ACI N

(addition with carry immediate — прямое сложение с учетом переноса); регистр общего назначения,

ADC R

(addition with carry — сложение с учетом переноса); содержимое ячейки памяти с адресом, хранящимся в HL,

82

ADC M

Результат выполнения — сумма соответствующего числа и содержимого аккумулятора, к которой прибавляется еще единица, если (C)=1. Результат записывается в аккумулятор. Влияние на регистр F такое же, как выше.

Рассмотрим примеры. Если (A)=22H, (HL)=0FFFFH, в ячейке памяти с адресом 0FFFFH хранится число 33H и (C)=1, то после выполнения операции ADC M в A будет число 56H. Если (A)=06H, (H)=0, а в результате какой-то прошлой операции флаг C был установлен в 1, то после выполнения операции ADC H в аккумуляторе окажется число 07H.

Перейдем теперь к вычитанию. Оно происходит аналогично: из содержимого аккумулятора вычитается некоторая величина, определяемая операндом команды, и результат сохраняется в аккумуляторе. Единственное изменение в терминологии касается переноса: в случае вычитания пользуются термином «заем».

26—28. Простое вычитание:

SUI N

(subtraction immediate — прямое вычитание);

SUB R

(subtraction — вычитание);

SUB M

Воздействие на биты Z, S, P регистра F, как в уже описанных командах, бит C устанавливается в 1, если в процессе вычитания был произведен заем из 7-го бита, и в 0, если не был.

Рассмотрим пример. Пусть (A)=66H, (B)=21H. В результате выполнения команд SUB B в A окажется число 45H.

29—31. Вычитание с учетом переноса:

SBI N

(subtraction with borrow immediate — прямое вычитание с учетом заема (переноса));

SBB R

(subtraction with borrow — вычитание с учетом заема);

SBB M

При выполнении этих команд из содержимого аккумулятора вычитается величина, определяемая операндом команды, а если флаг переноса установлен в 1, то вычитается еще 1. Воздействие на регистр F, как у предыдущих команд.

Пусть (A)=58H, (HL)=3210H, в ячейке памяти с адресом 3210H хранится число 42H, (C) в результате предыдущих действий равен 1. Тогда после выполнения команды SBB M в аккумуляторе окажется число 15H, содержимое ячейки памяти и HL останутся неизменными. А если (A)=32H, (B)=11H, (C)=1, то после выполнения операции SBB B в аккумуляторе окажется число 20H.

32—34. Команды сравнения вычитают из содержимого аккумулятора операнд (число; содержимое регистра общего назначения; содержимое ячейки памяти с адресом, находящимся в HL), но полученное число не записывается в аккумулятор. Таким образом, все операнды команд остаются неизменными, а результатом выполнения является изменение содержимого регистра F.

Мнемоники команд:

CFI N

(compare immediate — прямое сравнение);

CMF R

(compare — сравнивать);

CMF M

Содержимое регистра флагов меняется обычным образом:

Z устанавливается в 1, если результат вычитания равен 0, и в 0, если результат не равен 0;

S — в 1, если результат отрицательный, и в 0, если положительный;

P — в 1, если в битах результата четное число единиц, и в 0, если оно нечетное;

C — в 1, если в процессе вычитания был произведен заем из 7-го бита, и в 0, если не было.

Рассмотрим примеры. Пусть (A)=51H, (HL)=1234H, в ячейке памяти с адресом 1234H хранится число 52H. Тогда результатом выполнения операции CMF M будет установление следующих значений битов регистра F: (Z)=0, (S)=1, (P)=0, (C)=1. (A), (HL) и содержимое ячейки 1234H не изменяются.

Если (A)=63H, (B)=63H, то после выполнения команды CMF B флаги будут иметь

значения (Z)=1, (S)=0, (P)=0, (C)=0.

Настала очередь логических операций. Они оперируют с отдельными битами числа. Наиболее известные из них — AND (И), OR (ИЛИ). Результат логической операции определяется ее таблицей истинности. Приведем эти таблицы.

1 AND 1 = 1	1 OR 1 = 1
1 AND 0 = 0	1 OR 0 = 1
0 AND 1 = 0	0 OR 1 = 1
0 AND 0 = 0	0 OR 0 = 0

Еще одна логическая операция, реализованная в K580BM80,— XOR (исключающее ИЛИ). Это, по сути, сложение по модулю 2:

1 XOR 1 = 0
1 XOR 0 = 1
0 XOR 1 = 1
0 XOR 0 = 0

Выполнением логической операции с байтом считается ее выполнение с каждым из битов байта.

В качестве одного из операндов обязательно используется, как и в арифметических операциях, аккумулятор; в качестве другого операнда может выступать, как и прежде, явно заданное 8-разрядное число, либо содержимое регистра общего назначения, либо содержимое ячейки памяти, адресуемой с помощью HL.

ANI N

(AND immediate);

ANA R

ANA M

При выполнении этих команд происходит побитное логическое умножение (операция И) содержимого аккумулятора с числом, определяемым операндом. Результат записывается в аккумулятор. Флаг Z принимает значение 1, если результат равен 0, и 0, если не равен; S — 1, если результат отрицательный, 0 — если положительный; P и C — как сказано выше.

Пусть в аккумуляторе хранится число 99H=10111001B. Тогда при выполнении операции ANI 85H (85H=1000101B) мы получим результат 1000001B=81H, который и будет записан в аккумулятор.

38—40. Логическое сложение (OR):

ORI N

ORA R

ORA M

Команды выполняют побитное логическое сложение содержимого аккумулятора и операнда. Воздействие на флаги аналогично команде AND.

Пусть в аккумуляторе содержится число 82H, а в регистре C — 98H; тогда после выполнения операции ORA C в аккумуляторе окажется число 9AH.

41—43. «Исключающее ИЛИ»:

XRI N

XRA R

XRA M

Команды выполняют побитно операцию XOR между содержимым аккумулятора и операнда. Воздействие на флаги аналогично команде AND.

Пусть в аккумуляторе записано число 13H=00010011B, а в регистре D — число 25H=00100101B. Тогда после выполнения операции XRA D в аккумуляторе будет храниться число 36H=00110110B.

Перейдем теперь к довольно специфической группе арифметических операций — инкрементам и декрементам, иначе говоря, операциям увеличения или уменьшения на 1. Операндом в них может быть регистр общего назначения, либо ячейка памяти с адресом, хранящимся в HL, либо регистровая пара BC, DE, HL, либо регистр SP.

INR R

44—45. Увеличение 8-разрядного операнда на 1: (increment — увеличение);

INR M

N n/p	Мнемоника и длительность выполнения	Код и изменения биты регистра флагов	Действие
35	ANI N 7 тактов	11100110 NNNNNNNN Z, S, P, C, AC	Логическое умножение содержимого аккумулятора на 8-разрядное число
36	ANA R 4 такта	10100RRR Z, S, P, C, AC	Логическое умножение содержимого аккумулятора и регистра общего назначения
37	ANA M 7 тактов	10100110 Z, S, P, C, AC	Логическое умножение содержимого аккумулятора и ячейки памяти, адресуемой через HL
38	ORI N 7 тактов	11110110 NNNNNNNN Z, S, P, C, AC	Логическое сложение содержимого аккумулятора с 8-разрядным числом
39	ORA R 4 такта	10110RRR Z, S, P, C, AC	Логическое сложение содержимого аккумулятора и регистра общего назначения
40	ORA M 7 тактов	10110110 Z, S, P, C, AC	Логическое сложение содержимого аккумулятора и ячейки памяти, адресуемой через HL
41	XRI N 7 тактов	11101110 NNNNNNNN Z, S, P, C, AC	Операция "исключения или" над содержимым аккумулятора и 8-разрядным числом
42	XRA R 4 такта	10101RRR Z, S, P, C, AC	Операция "исключения или" над содержимым аккумулятора и регистра общего назначения
43	XRA M 7 тактов	10101110 Z, S, P, C, AC	Операция "исключения или" над содержимым аккумулятора и ячейки памяти, адресуемой через HL

35—37. Логическое умножение (AND):

На флаги эти операции воздействуют следующим образом:

Z устанавливается в 1, если результат равен 0, и в 0, если не равен;

S — в 1, если результат отрицательный, в 0 — если положительный;

P — в 1, если число битов результата, установленных в 1, четное, и в 0 — в противном случае;

C — не меняется.

Рассмотрим пример. Если в (A)=35H, то после выполнения операции INR A в A будет храниться число 36H.

46—47. Уменьшение 8-разрядного операнда на 1:

DCR R

(decrement — уменьшать);

DCR M

Действие на регистр флагов такое же, как в предыдущем случае.

84 Пусть (HL)=3333H, а в ячейке памяти с адресом 3333H хранится число 59H. Тогда после выполнения команды DCR M в ячейку памяти с адресом 3333H будет записано число 58H; содержимое регистровой пары HL не изменится.

Перейдем к командам 16-разрядной арифметики. Их всего три типа: инкремент, декремент и сложение (HL) с содержимым других регистровых пар.

48. Увеличение содержимого регистровой пары на 1:

INX RR

В коде команды

00RR0011

используемая регистровая пара обозначается следующим образом:

BC — 00

DE — 01

HL — 10

SP — 11

Таким образом, код команды INX H равен 00100011B=23H, и если в HL хранилось число 3321H, то после выполнения команды там будет записано число 3322H.

На флаги команда не воздействует.

49. Уменьшение содержимого регистровой пары на 1:

DCX RR

50. Сложение содержимого регистровой пары HL с одним из регистров BC, DE, HL, SP

DAD RR

Команда воздействует на флаг C. Он становится равным 1, если был перенос из 15-го бита, и 0, если переноса не было.

Рассмотрим в качестве примера сложение (HL) и (DE). Для этого выполним команду DAD D. Если в HL было записано число

1234H, а в DE — 4321H, то после выполнения команды в HL окажется число 5555H.

В арсенале вычислительных средств K580BM80 есть совсем уж необычные с точки зрения привычной арифметики команды: сдвиги. При сдвиге содержимое всех битов числа перемещается на одну позицию вправо или влево, так что при сдвиге вправо содержимое бита 1 попадает в бит 0, содержимое бита 2 — в бит 1 и т. д. Содержимое бита 0 попадает в бит C регистра F и, быть может, в бит 7; а может быть, в бит 7 попадет содержимое бита C. Это зависит от вида команды. Команды простого сдвига как бы сворачивают байт в кольцо. Перемещая содержимое битов на одну позицию вправо или влево, они еще и дублируют бит, переходящий через «место склейки» кольца, в бит C. Команды же сдвига через бит C «вставляют» бит C в «место склейки» кольца-байта, так что он служит не только приемником, но и источником информации.

Тест на понимание: объясните, почему сдвиг влево эквивалентен умножению байта на два, а сдвиг вправо — это, по сути, целочисленное деление на 2.

51—52. Сдвиг вправо содержимого аккумулятора. Таких команд две — простой сдвиг и сдвиг через бит C:

RRC

(rotate right circular — вращение вправо циклическое);

RAR

Пусть, для примера, (A)=10110111B, (C)=0. После выполнения команды RRC в аккумуляторе окажется число 11011011B, а C примет значение 1. А если бы мы выполнили команду RAR, то флаг C также принял бы значение 1, но в аккумуляторе оказалось бы число 01011011.

53—54. Сдвиг влево содержимого аккумулятора:

RLC

(rotate left circular — вращение влево циклическое);

RAL

Попробуем поворачивать аккумулятор, записав в него число 01110011, если флаг C равен 1. При выполнении команды RLC в аккумуляторе окажется число 11100110, (C)=0; при выполнении команды RAL в аккумуляторе окажется число 11100111, (C)=0.

Наконец, четыре последние команды из группы арифметических и логических.

55. Инвертирование флага C:

CMC

(complement carry — дополнение флага переноса). При выполнении этой команды C становится равным 0, если был равен 1, и 1,

если был равен 0. Старое значение С перепи-
сывается в бит AC регистра F.

56. Установка флага С в 1:

STC

(set carry — установить флаг переноса). При выполнении этой команды флаг С устанавливается в 1, а флаг AC — в 0.

57. Инвертирование содержимого аккумуля-
лятора:

CMA

(complement accumulator — дополнение ак-
кумулятора). Эта команда инвертирует все
биты аккумулятора. Если (A)=01100011,
после выполнения команды CMA в аккумуля-
торе окажется число 10011100.

58. Команда десятичной поправки, появив-
шаяся как результат существования двух
представлений двоичных чисел. Кроме обыч-
ного, с которым мы постоянно имеем дело,
существует еще двоично-десятичное пред-
ставление: цифрами 0 и 1 кодируются цифры
десятичного представления числа, а не само
число. Иначе говоря, байт разбивается на
4-разрядные половинки, в каждую из которых
разрешается записать двоичное число без
знака, не превышающее 9. Таким образом
можно кодировать десятичные числа от 0
до 99.

Чтобы перевести число из двоичного в
двоично-десятичное представление, нужно к
тем из его 4-разрядных половинок, содер-
жимое которых превышает 9, прибавить чис-
ло 6 (0110B). Разумеется, если возникает
перенос, то его тоже нужно прибавить. Эту
процедуру и выполняет команды.

DAA

(decimal adjust accumulator — добавление
к аккумулятору десятичного остатка). Опе-
рирует она, как следует из названия, с чис-
лом, находящимся в аккумуляторе.

Поскольку с двоично-десятичными (BCD —
binary-decimal digit) числами мы пока не
встречались, поупражняемся.

Допустим, мы хотим сложить числа 13 и
28. Результат известен — 41. Однако если
мы закодируем их в двоично-десятичном ви-
де (1D=0001B, 3D=0011B, 13D=
=00010011BCD; 2D=0010B, 8D=1000B,
28D=00101000BCD) и сложим, то...

```
00010011BCD
+
00101000BCD
-----
00111011B=3BH
```

3BH не является BCD-числом. Для при-
ведения его в «правильный» вид исполним
команду DAA, т. е. прибавим к каждой
4-разрядной половине байта, в которой запи-
сано число, превышающее 9, число 6. В нашем

N п/п	Мnemonic и длитель- ность вы- полнения	Код и изменение биты регистра флагов	Действие
44	INR R 3 тактов	0 0 R R R 1 0 0 Z, S, P, AC	Увеличение содержи- мого регистра на 1
45	INR M 10 тактов	0 0 1 1 0 1 0 0 Z, S, P, AC	Увеличение содержи- мого ячейки памяти, адресуемой через HL, на 1
46	DCR R 3 тактов	0 0 R R R 1 0 1 Z, S, P, AC	Уменьшение содержи- мого регистра на 1
47	DCR M 10 тактов	0 0 1 1 0 1 0 1 Z, S, P, AC	Уменьшение содержи- мого ячейки памяти, адресуемой через HL, на 1
48	INX RR 3 тактов	0 0 R R R 0 0 1 1	Увеличение содержи- мого регистровой па- ры на 1
49	DCX RR 3 тактов	0 0 R R R 1 0 1 1	Уменьшение содержи- мого регистровой па- ры на 1
50	DAD RR 10 тактов	0 0 R R R 1 0 0 1 C	Сложение содержи- мого HL с одной из регис- тровых пар
51	RRC 4 такта	0 0 0 0 1 1 1 1 C	Сдвиг содержимого аккумулятора вправо простой
52	RAR 4 такта	0 0 0 1 1 1 1 1 C	Сдвиг содержимого аккумулятора вправо через бит C
53	RLC 4 такта	0 0 0 0 0 1 1 1 C	Сдвиг содержимого аккумулятора влево простой
54	RAL 4 такта	0 0 0 1 0 1 1 1 C	Сдвиг содержимого аккумулятора влево через бит C
55	CMC 4 такта	0 0 1 1 1 1 1 1 C	Инвертирование флага C
56	STC 4 такта	0 0 1 1 0 1 1 1 C	Установка флага С в 1
57	CMA 4 такта	0 0 1 0 1 1 1 1	Инвертирование со- держимого аккумуля- тора
58	DAA 4 такта	0 0 1 0 0 1 1 1 Z, S, P, C, AC	Приведение двоичного числа к двоично-де- сятичному представ- лению

случае необходимо прибавить 6 к «младшей»
половине:

```
00111011B
+
00000110B
-----
01000001BCD=41D
```

Результат правильный.

Эта команда воздействует на флаги сле-
дующим образом:

Z устанавливается в 1, если в результате
получился 0, и в 0 — в противном случае;
в S записывается содержимое старшего
бита аккумулятора;

P устанавливается в 1, если число единиц
в байте четное, и в 0, если нечетное;

C — в 1, если был перенос из старшего
бита, и в 0, если не было.

Возможно, набор вычислительных команд
покажется читателю очень скудным. Однако
его достаточно, чтобы проделать все матема-
тические вычисления, в том числе и с дей-
ствительными числами. Правда, каждое такое
вычисление производится при помощи огром-
ного числа разобранных нами простейших
команд, поэтому скорость вычислений с пла-
вающей запятой оказывается довольно не-
высокой.

Продолжение следует.

Архитектура «Корвета»

Следующий пример показывает рисование горизонтальных и вертикальных линий. Обратите внимание на разницу в скорости их вывода.

```
(пример 13 , линии)
program Lines;
Const
  GrRamBase      = $4000;
  (начало видеопамати в конфигурации 1CH)
  SysRegister    = $FA7F;
  ( адрес системного регистра )
  ColorRegister  = $FABF;
  ( регистра цвета )
  SysGrRegister  = $FF7F;
  (адрес системного регистра в
   конфигурации $3C)
  SysConfig      = $1C;
  (конфигурация CP/M)
  GraphConfig    = $3C;
  (конфигурация при включенной графике)
  GraphLine      = 64;
  (длина графической гориз. линии)
  ESC            = #27;
  SPACE         = #32;
  iDI           = $F3;
  iEI           = $FB;
  Convert:array[0..7] of byte
    = ($80,$40,$20,$10,8,4,2,1);
  (для преобразования номера точки )

var
  Addr : integer;
  x,y : integer; (для координат)
  Color: byte;
  Mask : byte;

begin
  ClrScr;
  ClrGScr;
  Color := 0;
  Addr := GrRamBase;
  GotoXY(1,15);
  Write(' горизонтальные линии... ');
  for y := 0 to 240 do
    begin
      mem[ColorRegister] := $80 OR
        ((Color and 7) shl 1); {цвет}
      inline (iDI);
      ( запретили прерывания )
      mem[SysRegister] := GraphConfig;
      for x := 0 to 63 do
        mem[Addr+x] := $FF;
      (пишем сразу 8 точек)
      mem[SysGrRegister] := SysConfig;
      inline (iEI);
      ( разрешили прерывания )
      Inc (Addr, GraphLine);
      ( следующая строка )
      Inc (Color);
      ( сменим цвет )
    end;
  Write ('      нажмите ПРОБЕЛ');
  repeat until keypressed;
  ClrScr;
  ClrGScr;
  Color := 0;
```

```
Addr := GrRamBase;
GotoXY(1,15);
Write('
      вертикальные линии... ');
for x := 0 to 511 do
  begin
    mem[ColorRegister] := $80 OR
      ((Color and 7) shl 1); {цвет}
    inline (iDI);
    ( запретили прерывания )
    mem[SysRegister] := GraphConfig;
    Mask := Convert[x and 7];
    Addr := GrRamBase + (x shr 3);
    Inc (Addr, GraphLine);
    ( следующая строка )
    for y := 0 to 240 do
      begin
        mem[Addr] := Mask;
        Inc (Addr, GraphLine);
        ( следующая строка )
      end;
    mem[SysGrRegister] := SysConfig;
    inline (iEI);
    ( разрешили прерывания )
    Inc (Color);
    ( сменим цвет )
  end;
end.
```

Предыдущие примеры показывают, как можно записывать в ГЗУ в режиме цвета. Теперь попробуем читать в этом же режиме. Напомним, что при чтении происходит аппаратное сравнение двух цветов — заданного в регистре цвета и цвета точки на экране.

Чтобы лучше разобраться, что происходит при чтении, выведем в левом верхнем углу экрана линию — 8 горизонтальных точек (т. е. запишем соответствующие данные в байт с адресом 0 в ГЗУ), каждая со своим цветом, — и будем читать нулевой байт, меняя лишь цвет чтения (биты 4—6 регистра цвета) после каждого нажатия клавиши ПРОБЕЛ. Результат чтения для наглядности представляется на экране в двоичном виде.

```
(пример 14 , чтение в режиме ЦВЕТ)
program ReadColor;
Const
```

```
GrRamBase      = $4000;
SysRegister    = $FA7F;
ColorRegister  = $FABF;
SysGrRegister  = $FF7F;
SysConfig      = $1C;
GraphConfig    = $3C;
GraphLine      = 64;
ESC            = #27;
SPACE         = #32;
iDI           = $F3;
iEI           = $FB;
Convert:array[0..7] of byte
  = ($80,$40,$20,$10,8,4,2,1);
```

```
var
  Addr : integer;
  x,y : integer;
  Color: byte;
```

Окончание. Начало см.: Информатика и образование. 1992. № 1—3, 4.

86


```

Mask : byte;
ReadByte : byte;
Ch : char;

procedure
  WritePoint(x,y,Color:integer);
begin
  Mask := Convert[(x and 7)];
  Addr := GrRamBase + y*64 +
           (x shr 3);
  mem[ColorRegister] := $80 OR
    ((Color and 7) shl 1); {цвет}
  inline (iDI);
  { запретили прерывания }
  mem[SysRegister] := GraphConfig;
  mem[Addr] := Mask;
  mem[SysGrRegister] := SysConfig;
  inline (iEI);
  { разрешили прерывания }
end;

procedure WriteBin(Data:byte);
var i:integer;
begin
  i := 128;
  repeat
    if (i and Data) = i then Write('1')
    else Write('0');
    i := i shr 1;
  until i = 0;
end;

begin
  ClrScr;
  ClrGScr;
  Color := 0;
  for x := 0 to 7 do
    WritePoint(x,0,x);
  Color := 0;
  repeat
    mem[ColorRegister] := $80 or
      ((Color and 7) shl 4); {цвет чтения}
    GotoXY(1,15);
    Write('Цвет чтения - ',Color
      and 7,'; читаем - ');
    inline (iDI);
  { запретили прерывания }
  mem[SysRegister] := GraphConfig;
  ReadByte := mem[GrRamBase];
  mem[SysGrRegister] := SysConfig;
  inline (iEI);
  { разрешили прерывания }
  WriteBin(ReadByte);
  Inc(Color);
  Ch := ReadKey;
  until Ch = ESC;
end.

```

Теперь разберемся с другим режимом обращения к ГЗУ — плоскостным. Он называется так, поскольку дает возможность обращаться отдельно к каждой плоскости ГЗУ. Такая необходимость возникает, например, если мы хотим записать сюда какие-либо числовые данные. Следующий пример показывает, как в этом режиме следует обращаться к графической памяти.

(пример 15, чтение/запись в режиме ПЛОСКОСТЬ)

```

program PlaneAccess;
Const
  GrRamBase = $4000;
  SysRegister = $FA7F;

```

```

{ адрес системного регистра }
ColorRegister = $FABF;
{ регистра цвета }
ColorGrRegister=$FFBF;
{ регистра цвета }
SysGrRegister = $FF7F;
SysConfig = $1C;
GraphConfig = $3C;
GraphLine = 64;
ESC = #27;
SPACE = #32;
iDI = $F3;
iEI = $FB;
ConvertWrite:array[0..2] of
  byte = ($0C,$0A,$06);
ConvertRead:array[0..2] of
  byte = ($10,$20,$40);
{для преобразования номера плоскости
в значение регистра цвета}
TextPlane:array [0..2] of
  string[30] =
('запись/чтение в плоскость 0',
'запись/чтение в плоскость 1',
'записи/чтение в плоскость 2');
var
  Address : integer;
  Plane: byte;
  Ch : char;
  i : integer;

```

87

```

procedure WriteByte(Address,Plane:
  integer;Ch:Char);
begin
  mem[ColorRegister] :=
    ConvertWrite[Plane];
  inline (iDI);
  { запретили прерывания }
  mem[SysRegister] := GraphConfig;
  mem[Address] := $FF;
  { обнуляем ячейку графической памяти }
  mem[ColorGrRegister] :=
    ConvertWrite[Plane] or 1;
  {для записи данных}
  mem[Address] := Ord(Ch);
  { записываем данные в ячейку
  графической памяти }
  mem[SysGrRegister] := SysConfig;
  inline (iEI);
  { разрешили прерывания }
end;

function ReadByte(Address,Plane:
  integer):char;
begin
  mem[ColorRegister] :=
    ConvertRead[Plane];
  { установили плоскость }
  inline (iDI);
  { запретили прерывания }
  mem[SysRegister] := GraphConfig;
  { включили графическую память }
  ReadByte := chr(mem[Address]);
  { читаем ячейку графической памяти }
  mem[SysGrRegister] := SysConfig;
  inline (iEI);
  { разрешили прерывания }
end;

```

```

begin
  ClrScr;
  ClrGScr;
  Address := GrRamBase;
  for Plane := 0 to 2 do
    begin
      for i := 1 to
        length(TextPlane[Plane]) do

```

```

begin
  WriteByte(Address, Plane,
    TextPlane[Plane][i]);
  delay(100); {для красоты}
  Inc(Address);
end;
end;
ClrScr;
GotoXY(0,3);
Address := GrRamBase;
for Plane := 0 to 2 do
begin
  repeat
    ch := ReadByte(Address, Plane);
    if ch <> #0 then
      begin
        write(ch);
        delay(100); {для красоты}
        Inc(Address);
      end;
    until ch = #0;
    WriteLn;
  end;
end.

```

88 Кстати, а сколько у вас графической памяти? Если, запустив следующую программу, вы увидите на экране четыре горизонтальные линии, значит, вам не повезло — в вашем «Корвете» всего 48 Кбайт ГЗУ. Если же на экране только одна линия — нажимайте **ПРОБЕЛ**: изображения четырех линий записаны в четыре страницы ГЗУ, и вы увидите все четыре по очереди.

{пример 16}

```

program TestScreen;
const

```

```

  VideoRegister = $FB3A;
  SPACE = #32;
  ESC = #27;
  GrRamBase = $4000;
  SysRegister = $FA7F;
  ColorRegister = $FABF;
  SysGrRegister = $FF7F;
  SysConfig = $1C;
  GraphConfig = $3C;
  GraphLine = 64;
  iDI = $F3;
  iEI = $FB;

```

```

var
  Address : integer;
  x, y : integer;
  Page : byte;
  Ch : char;

```

```

procedure RowLine(y:integer);
begin

```

```

  inline (iDI);
  { запретили прерывания }
  mem[SysRegister] := GraphConfig;
  for x := 0 to 63 do
    mem[GrRamBase+64*y+x] := $FF;
  {пишем сразу 8 точек}
  mem[SysGrRegister] := SysConfig;
  inline (iEI);
  { разрешили прерывания }
end;

```

```

procedure pcls;
begin
  Address := GrRamBase;
  for y := 0 to 255 do
    begin

```

```

      mem[ColorRegister] := $80 ;
      {цвет черный}
      inline (iDI);
      { запретили прерывания }
      mem[SysRegister] := GraphConfig;
      for x := 0 to 63 do
        mem[Address+x] := $FF;
      {пишем сразу 8 точек}
      mem[SysGrRegister] := SysConfig;
      inline (iEI);
      { разрешили прерывания }
      Inc(Address, GraphLine);
      { следующая строка }
    end;
  end;
begin
  ClrScr;
  ClrGScr;
  Write('Идет процесс... Ж Д И Т Е');
  mem[VideoRegister] :=
    (mem[VideoRegister] or $C0);
  Pcls;
  mem[ColorRegister] := $8E ;
  {цвет белый}
  RowLine(50);
  mem[VideoRegister] :=
    (mem[VideoRegister] and $3F) or $80;
  Pcls;
  mem[ColorRegister] := $8E ;
  {цвет белый}
  RowLine(100);
  mem[VideoRegister] :=
    (mem[VideoRegister] and $3F) or $40;
  Pcls;
  mem[ColorRegister] := $8E ;
  {цвет белый}
  RowLine(150);
  mem[VideoRegister] :=
    (mem[VideoRegister] and $3F);
  Pcls;
  mem[ColorRegister] := $8E ;
  {цвет белый}
  RowLine(200);
  ClrScr;
  Write('Теперь жмите ПРОБЕЛ');
  Page := 0;
  repeat
    Ch := ReadKey;
    if Ch = SPACE then
      begin
        mem[VideoRegister] :=
          (mem[VideoRegister] and $FC)
            or (Page and 3);
        Inc(Page);
      end;
    until Ch = ESC;
    mem[VideoRegister] :=
      (mem[VideoRegister] and $FC);
    { восстановим страницу 0 }
  end.

```

Следующая демонстрация, «бегущие прямоугольники», покажет пользу просмотрной таблицы и технику обращения к ней. Поскольку такое обращение не требует переключения карты памяти, этот пример можно переписать на Бейсике.

{пример 17, просмотрная таблица}

```

program Lut;
Const
  LutRegister = $FAFB;
  { адрес просмотрной таблицы }

```

```

var      Offset : integer;
         i : integer;
{ для координат }
         Color: byte;

begin
  ClrScr;
  ClrGScr;
  Color := 7;
  for i := 0 to 125 do
    {заполняем поле прямоугольниками}
    begin
      SetColor(Color);
      Rectangle(254-(i+i),126-i,
                258+i,i,128+i,false);
      Color := (Color - 1) ;
    { сменим цвет }
      if Color = 0 then Color := 7;
    end;
    Offset := 0;
    repeat
      for i := 1 to 7 do
        mem[LutRegister] := i or
          ((Offset + i) AND 7) shl 4);
        delay(20);
        Inc(Offset);
      until keypressed;
    end.

```

Запустив программу, можно видеть, что эффект «бегущих прямоугольников» сопровождается помехами на экране. Это следствие нарушения отображения на экране в момент обращения к просмотровой таблице.

Избежать нарушения можно, обращаясь к просмотровой таблице только в то время, когда отображение электронного луча запрещено, например во время обратного хода кадровой развертки. В следующей модификации данной программы этот недостаток устранен.

```

{пример 18, просмотровая таблица}
program LutGood;
Const
  LutRegister = $FAFB;
  Reg3BH = $FB3B;
  BlankBit = 2;
  DataLut:array [1..13] of byte =
    ($10,$20,$30,$40,$50,$60,$70,$10,
     $20,$30,$40,$50,$60);

```

```

var      ValDelay : integer;
{величина задержки}
         Offset : byte;
         i : byte;
{ для координат }
         Color: byte;

Function TestBlank:boolean;
{ проверяет вертикальный шланк }
begin
  TestBlank := (mem[Reg3BH]
                AND BlankBit) = 0;
end;
procedure Del(count:integer);
var i:integer;
begin
  for i := 0 to count do;
end;
procedure WaitBlank;
begin

```

```

repeat
until Not TestBlank;
repeat
until TestBlank;
end;

begin
  ClrScr;
  ClrGScr;
  Color := 7;
  for i := 0 to 125 do
    {заполняем поле прямоугольниками}
    begin
      SetColor(Color);
      Rectangle(254-(i+i),126-i,
                258+i,i,128+i,false);
      Color := (Color - 1) ;
    { сменим цвет }
      if Color = 0 then Color := 7;
    end;
    Offset := 0;
    repeat
      WaitBlank;
      Del(10); { эта задержка дает
                тонкую регулировку, для уточнения
                обратного хода луча }
      for i := 1 to 7 do
        mem[LutRegister] :=
          i or DataLut[Offset+i];
        Offset := (Offset +1) AND 7;
      until keypressed;
    end.

```

89

В завершение описания графического дисплея покажем, как можно нетрадиционно использовать возможности процессора для повышения эффективности программы. Ниже приводятся две программы на ассемблере, выполняющие закрашивание экрана. Наберите их, хотя бы в отладчике, запустите (обе закрашивают экран заданным цветом) и разберитесь, почему так различается их эффективность.

```

PIA1 EQU 0fb3BH ;адрес порта
                      статуса принтера
GrConfig EQU 3CH
SysReg EQU 0FA7FH
SysGrReg EQU 0FF7FH
ColReg EQU 0FFBFH
StartGraph EQU 4000H
BDOS EQU 5

```

```

START:
CALL PCLS
LDA COLOR
INR A
ANI 7
STA COLOR ;следующий цвет
MVI C,6 ;функция BDOS -
          прямое обращение к консоли
MVI E,0FFH ;чтение статуса
CALL BDOS
ORA A ;проверяем
          нажатие клавиши
JZ START ;никакая
          клавиша не нажата
RET ;эти две
RST 7 ;строочки надо поменять местами,
          ;если провозать
          под отладчиком

PCLS:
DI ;запретим
          прерывания

```

```

MVI A,GrConfig
STA SysReg ;подключим
                графику
LXI H,StartGraph
LXI B,4000H ;счетчик
LDA COLOR ;регистр цвета
ADD A ;удвоили
ORI 80H ;режим цвета
STA ColReg
LPCLS1: MVI M,0FFH
INX H
DCR C ;младший байт
                счетчика
JNZ LPCLS1
DCR B ;старший байт
                счетчика
JNZ LPCLS1
MVI A,1CH
STA SysGrReg ;восстановим
                системный региср
EI ;разрешили
                прерывания
RET ;закончили
COLOR: DB 0 ;здесь хранится
                текущий цвет
END
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PIA1 EQU 0fb30H ;адрес порта
                статуса принтера
GrConfig EQU 3CH
SysReg EQU 0FA7FH
SysGrReg EQU 0FF7FH
ColReg EQU 0FFBFH
EndGraph EQU 8000H
BDOS EQU 5
START: CALL PCLS
LDA COLOR
INR A
ANI 7
STA COLOR ;следующий цвет
MVI C,6 ;функция BDOS -
                прямое обращение к консоли
MVI E,0FFH ;чтение статуса
CALL BDOS
ORA A ;проверяем
                нажатие клавиши
JZ START ;никакая
                клавиша не нажата
RET ;эти две строчки
                надо поменять местами, если
                RST 7 ;пробовать под
                отладчиком
PCLS: LXI H,0
DAD SP
XCHG
DI ;запретим
                прерывания
MVI A,GrConfig
STA SysReg ;подключим
                графику
LXI SP,EndGraph
LXI H,0FFFFH ;маска записи
LXI B,4000H ;счетчик
LDA COLOR ;регистр цвета
ADD A ;удвоили
ORI 80H ;режим цвета
STA ColReg
LPCLS1: PUSH H
PUSH H
PUSH H
PUSH H
PUSH H ;быстро-быстро
                записываем 16 байт
                в графическую
                память
PUSH H
DCR C ;младший байт
                счетчика
JNZ LPCLS1 ;старший байт
                счетчика
DCR B
JNZ LPCLS1
XCHG
SPHL ;восстановим
                указатель стека
MVI A,1CH
STA SysGrReg ;восстановим
                системный региср
EI ;разрешили
                прерывания
RET ;закончили
COLOR: DB 0 ;здесь хранится
                текущий цвет
END

```

На этом завершено описание дисплея «Корвета». Теперь вы знакомы со всеми тонкостями работы с этими устройствами.

А. ДМИТРИЕВ

«Корвет» 2 → 1

Отличия второй версии ПЗУ «Корвета» от первой привели к программной несовместимости компьютеров разных выпусков. Видимо, каждому обладателю «Корвета» 2.0 знакомо чувство разочарования, когда с трудом добытые программы отказываются нормально работать только потому, что они были написаны для более ранней модификации машины.

Между тем адаптация Бейсик-программ ко второй версии ПЗУ достаточно проста. Достаточно знать, что в ПЗУ-2 изменен принцип обработки сигналов драйвером клавиатуры, имеются существенные отличия по организации вывода на экран расширенных символов и, наконец, иначе кодируются прерывания.

Прежде всего надо переписать програм-

му на диск с новой, «родной» для ПЗУ-2 версией МикроДОС. Для копирования следует использовать COPY (COPYT, COPYD и им подобные вместе с программами скопируют и старую МикроДОС).

Прерываниями при работе с Бейсиком заведует ячейка FB29. В большинстве программ она используется, во-первых, для отключения компьютера от локальной сети. Для этого где-то в начале программы может стоять оператор

```
POKE &HFB29,147
```

В новой версии он отключает и клавиатуру. Исправьте его, написав

```
POKE &HFB29,8
```

или просто удалите — и программа начнет работать. Во-вторых, в ряде контролирующих программ оператор

```
POKE &HFB29,255,
```

стоящий после вывода результата, блокирует клавиатуру, чтобы ученик не мог скрыть свою оценку от учителя. Найдите и удалите его, по крайней мере на время отладки программы. Иначе компьютер может зависнуть и вам не удастся сохранить исправленную программу на диске.

При использовании ПЗУ-1 режим вывода на экран расширенных символов включает-ся оператором

```
PRINT CHR$(27)+"3"
```

При этом по-прежнему считается, что экран имеет 64 позиции по ширине, но выводятся на экран только символы, попадающие в четные позиции. Выключаются расширенные символы командой

```
PRINT CHR$(27)+"2"
```

На «Корвете» с ПЗУ-2 при включении режима расширенных символов ширина экрана устанавливается равной 32 позициям и расширенный текст, написанный для старых машин, искажается до неузнаваемости. Для исправления положения достаточно включить режим расширенных символов после вывода текста на экран, а не до. Если, например, в исходной программе есть фрагмент типа

```
100 PRINT CHR$(27)+"3"
110 PRINT "PFAACSSHIIIPPEENNYEE
          CСИИМВВООЛЛЫ"
120 PRINT "KКооррввеетт---ввееерссииияя
          11..00"
```

то его нужно исправить следующим образом:

```
110 PRINT "PFAACSSHIIIPPEENNYEE
          CСИИМВВООЛЛЫ"
120 PRINT "KКооррввеетт---ввееерссииияя
          11..00"
121 PRINT CHR$(27)+"3"
```

К сожалению, этот прием срывает не всегда, поэтому бывает проще переписать такие фрагменты программы заново, используя для вывода расширенных симво-

лов операторы SCREEN,,1 и SCREEN,,0.

В старой версии для определения кода нажатой клавиши использовалось прямое сканирование клавиатуры; код нажатой клавиши записывался в ячейку F701. Большинство программ для управления ходом программы использовало конструкцию вида

```
150 S=PEEK(&HF701): IF S=0 THEN 150
160 IF S=67 THEN GOTO 1000
170 IF S=73 THEN GOTO 2000
```

Для ПЗУ-2 придется заменить строку 150:

```
150 S*=INKEY$: IF S*="" THEN 150
          ELSE S=ASC(S*)
```

Это еще не все. Дело в том, что ASCII-коды (коды КОИ-8) и скан-коды клавиш не совпадают, поэтому необходимо изменить и строки 160—170:

```
160 IF S=25 THEN GOTO 1000
170 IF S=26 THEN GOTO 2000
```

Скан-коды и коды КОИ-8 наиболее часто используемых клавиш приведены в табл. 1. Остальные скан-коды клавиш можно найти в книге № 4 технического описания («Системное программное обеспечение», приложение 13), а коды КОИ-8 — в книге № 6 («Интерпретатор языка Бейсик», приложение 1). Группа клавиш режима клавиатуры (УПР, РГ, АЛФ, ГРАФ, СЕЛ) не имеет собственных ASCII-кодов. Определить, какая из них нажата в текущий момент, можно с помощью ячейки F880 из поля клавиатуры (табл. 2).

На «Корветах» разных лет выпуска могут различаться коды некоторых клавиш, поэтому бывает очень трудно определить,

Таблица 1

Клавиша	Код КОИ-8	Скан-код
БК (CR)	13	49
СТРП (CLS)	31	50
СТОП	3	51
ИЗ (DEL)	29	52
ВЗ (INS)	28	53
<= (BS)	8	54
TAB	9	55
Провал	32	56
ПРФ (ESC)	27	60
Клавиши дополнительного поля		
2	67	67
4	8	69
5 (МЕНЮ)	16	70
6	24	71
8	25	73
Клавиши режима клавиатуры		
РГ (левый)		57
АЛФ (ALF)		58
ГРАФ (GRAPH)		59
СЕЛ (SEL)		61
УПР (CTRL)		62
РГ (правый)		64

Таблица 2

PEEK (&HFB29)	Нажата клавиша
1 (00000001B)	FG (левая)
2 (00000010B)	АЛФ (ALF)
4 (00000100B)	ГРАФ (GRAPH)
16 (00010000B)	СЕЛ (SEL)
32 (00100000B)	УПР (CTRL)
128 (10000000B)	FG (правая)

какую клавишу имели в виду авторы программы в том или ином случае. Иногда для этого необходим тщательный анализ текста программы.

Вот, пожалуй, и все. Осталось только сохранить исправленную программу на диске.

Комментарий специалиста

Прерывания требуют особо осторожного обращения, поэтому для предотвращения отключения клавиатуры следует:

считать содержимое ячейки FB29;

запомнить его;

записать в FB29, заменив только нужный бит;

после окончания восстановить содержимое FB29.

Делается это следующим образом.

```
MEMMASC=PEEK (&HFB29)
POKE &HFB29, MEMMASC OR 8
.....
POKE &HFB29, MEMMASC
```

А. СКУРИХИН

Вниманию преподавателей информатики и физики!

Ваши ученики с нашей помощью приобретут прочные практические навыки в решении задач по физике и программированию на языке Бейсик.

За доступную цену предлагаем «Электронный задачник».

52 урока по методике решения задач по физике и 260 задач по программированию раскрывают 9 основных тем.

Наша оригинальная разработка — к Вашим услугам!

Необходимую информацию можно получить по телефонам: (06452) 4-01-46, 2-85-32.

Наш адрес: 349940, г. Северодонецк, Луганская обл., ул. Парижской Коммуны, 46. НПО «Квантор».

РОСЦИО

Российский центр информатизации образования



извещает
о выходе в ноябре 1992 г. первого номера каталога
«Компьютерные учебные программы».

Программисты и разработчики!

РОСЦИО напоминает:

65 000 школ, лицеев, колледжей России, сотни ПТУ, училищ, педвузов России ждут сообщений о Ваших разработках, опубликованных в каталоге.

Руководители учебных заведений России!

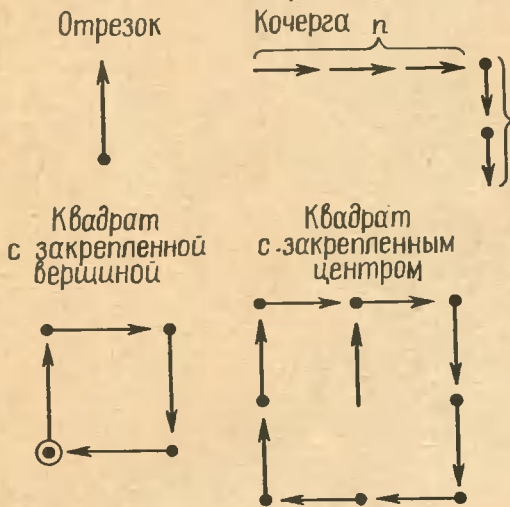
в каталоге «Компьютерные учебные программы» Вы найдете самые последние сведения об учебных программах и условиях их поставки.

Контактный телефон: (095) 924-61-97

М. СТЕПАНОВ, В. САМОЛЫСОВ

Дополнительные возможности графики ПЭВМ «Агат»

Выделим четыре типа образов (фигур): отрезок, кочерга, квадрат с закрепленной вершиной, квадрат с закрепленным центром (рис. 7).



Для каждой фигуры напишем программу, которая загружает таблицу с информацией в память компьютера, а затем запишем ее на диск.

```

10 REM ОТРЕЗОК
20 DATA 1,0,4,0,4,0
30 POKE REB,0:POKE RE9,810
40 FOR AD=81000 TO 81005
50 READ B
60 POKE AD,B
70 NEXT AD
80 NEW
    
```

Запишем программу на диск под именем ОТРЕЗОК: SAVE ОТРЕЗОК.

Перед использованием этой фигуры можно либо сразу загрузить таблицу в память: RUN ОТРЕЗОК, либо загрузить в память текст программы ОТРЕЗОК: LOAD ОТРЕЗОК и удалить 80-ю строку. Затем набирается новая программа или дополняется новыми строками загруженная в память про-

грамма ОТРЕЗОК. Готовая программа позволит манипулировать отрезком.

После того как проведена подготовительная работа, оператор DRAW можно рассматривать как команду, предназначенную для построения геометрической фигуры, поэтому наиболее естественными являются следующие постановки задач, решаемых с помощью этого оператора:

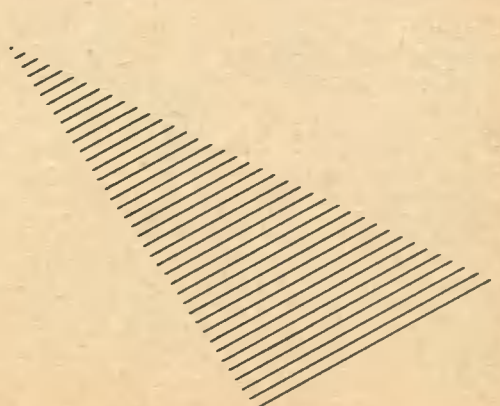
- 1) построить семейство геометрических фигур;
- 2) написать программу, которая заставит фигуру перемещаться.

Поскольку операторы ROT и SCALE имеют наглядную интерпретацию (вращение и рост фигуры), то задачи о перемещении фигур зачастую формулируются наиболее естественным образом. При этом в задаче может предполагаться, что на экране сохраняются все фазы движения фигуры. Ряд интересных задач возникает благодаря тому, что перед построением следующей фазы движения можно частично затереть предыдущую фазу.

Далее будут даны общие формулировки задач и приведены конкретные примеры на каждую формулировку.

Построить семейство геометрических фигур, представленное в задании визуально.

Пример 1. Постройте семейство отрезков, изображенное на рисунке 8.



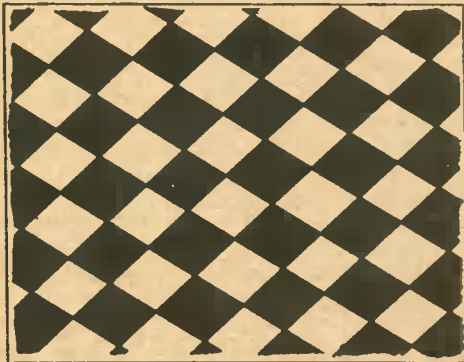
Окончание. Начало см.: Информатика и образование. 1992. № 3.

```

10 DATA 1,0,4,0,4,0
20 POKE #E8,0
30 POKE #E9,#10
40 FOR I=0 TO 5
50 READ A
60 POKE #1000+I,A
70 NEXT I
80 HGR=1:COLOR=15:ROT=40
90 FOR X=0 TO 255 STEP 5
100 SCALE=2*X/5
110 DRAW 1 AT X,X*4/5
120 NEXT X
130 GET FR:TEXT=0

```

Пример 2. Построить изображение ткани «домино» (рис. 9), представляющее собой семейство ромбов.



Построение проведем на базе квадрата с закрепленной вершиной

```

10 DATA 1,0,4,0,44,62,0
20 POKE #E8,0
30 POKE #E9,#10
40 FOR I=#1000 TO #1006
50 READ A
60 POKE I,A
70 NEXT I
80 HGR=1:COLOR=15:ROT=8
90 FOR X=0 TO 240 STEP 16
100 FOR Y=0 TO 240 STEP 16
110 FOR J=1 TO 9 STEP 2
120 SCALE=J
130 DRAW 1 AT X+11-J,Y
140 NEXT J
150 NEXT Y,X
160 GET FR:TEXT=0

```

Осуществить перемещение фигуры без сохранения промежуточных фаз движения.

Пример. Напишите программу, которая «заставит катиться» квадрат с закрепленным центром.

```

10 DATA 1,0,4,0,40,54,63,36,5,0
20 POKE #E8,0
30 POKE #E9,#10
40 FOR I=#1000 TO #1009
50 READ A
60 POKE I,A
70 NEXT I
80 HGR=1:COLOR=15
90 FOR X=0 TO 255
100 ROT=X
110 XDRAW 1 AT X,120
120 FOR T=1 TO 20:NEXT T

```

```

130 XDRAW 1 AT X,120
140 NEXT X
150 GET FR:TEXT=0

```

Осуществить перемещение фигуры с сохранением промежуточных фаз движения.

Пример 1. Напишите программу, которая нарисует морского ежа с помощью вращающегося отрезка постоянной длины.

```

10 DATA 1,0,4,0,4,0
20 POKE #E8,0
30 POKE #E9,#10
40 FOR I=#1000 TO #1005
50 READ A
60 POKE I,A
70 NEXT I
80 HGR=1:COLOR=15
90 FOR U=0 TO 64
100 SCALE=90:ROT=U
110 DRAW 1 AT 127,127
120 NEXT U
130 GET FR:TEXT=0

```

Пример 2. Напишите программу, которая нарисует бурв с помощью растущего и одновременно вращающегося отрезка, конец которого скользит по прямой.

```

10 DATA 1,0,4,0,4,0
20 POKE #E8,0
30 POKE #E9,#10
40 FOR I=#1000 TO #1005
50 READ A
60 POKE I,A
70 NEXT I
80 HGR=1:COLOR=15
90 FOR Y=0 TO 255
100 SCALE=Y/4:ROT=Y
110 DRAW 1 AT 100+Y/4,Y
120 NEXT Y
130 GET FR:TEXT=0

```

Пример 3. Заменяв в предыдущей программе «отрезок» на «кочергу 1,1», нарисуйте «анаконду».

Заменяем строку

```
10 DATA 1,0,4,0,51,0
```

Пример 4. Напишите программу, которая изобразит фазы движения вращающегося отрезка постоянной длины, конец которого скользит по окружности.

```

10 DATA 1,0,4,0,4,0
20 POKE #E8,0
30 POKE #E9,#10
40 FOR I=#1000 TO #1005
50 READ A
60 POKE I,A
70 NEXT I
80 HGR=1:COLOR=15
90 FOR U=0 TO 6.4 STEP .03
100 SCALE=20:ROT=U*40
110 X=127+80*X SIN(U):Y=127-80*Y COS(U)
120 DRAW 1 AT X,Y
130 NEXT U
140 GET FR:TEXT=0

```


Построить изображение с помощью частичного сохранения фаз движения фигуры.

Пример. Построить эллипс небольшого радиуса, затира фазы движения вращающегося отрезка за исключением одной точки.

```
10 DATA 1,0,4,0,4,0
20 POKE MEM,0
30 POKE ME9,110
40 FOR I=11000 TO 11005
50 READ A
60 POKE I,A
70 NEXT I
80 HGR=1:COLOR=15
90 FOR U=0 TO 63
100 ROT=U
110 SCALE=10:DRAW 1 AT 127,127
120 SCALE=9:XDRAW 1 AT 127,127
130 NEXT U
140 GET FR:TEXT=0
```

Сочетая различные приемы использования операторов DRAW и XDRAW, достичь интересного эффекта.

Пример 1. Напишите программу движения секундной стрелки по циферблату.

```
10 DATA 1,0,4,0,4,0
20 POKE MEM,0:POKE ME9,115
30 FOR I=0 TO 5
40 READ A
50 POKE 11500+I,A
60 NEXT I
70 HGR=1:COLOR=15
80 FOR U=0 TO 63
90 ROT=U:SCALE=100:DRAW 1 AT 127,127
100 SCALE=85:XDRAW 1 AT 127,127
110 NEXT U
120 SCALE=80
130 FOR U=0 TO 63
140 ROT=U
150 XDRAW 1 AT 127,127
160 FOR T=1 TO 600:NEXT T
170 PRINT CHR(7)
180 XDRAW 1 AT 127,127
190 NEXT U
200 GOTO 130
```

Пример 2. Нарисуйте на экране забавную рожицу.

Данная программа может рассматриваться как пример работы с несколькими образами одновременно

```
10 DATA 3,0,8,0,12,0,14,0
20 DATA 219,219,6,0:REM КОЧЕРГА
30 DATA 4,0 :REM ОТРЕЗОК
40 DATA 58,36,45,54,7,0:REM КВАДРАТ
50 POKE ME9,113:POKE MEM,0
60 FOR I=0 TO 19
70 READ A
80 POKE 11500+I,A
90 NEXT I:HGR=2:COLOR=15
100 REM PABOTA C 1-M ОБРАЗОМ
110 SCALE=20
120 FOR U=0 TO 63
130 ROT=U:DRAW 1 AT 127,127
140 NEXT U
150 REM PABOTA CO 2-M ОБРАЗОМ
160 X=80:Y=80
170 FOR I=1 TO 2
180 FOR U=0 TO 63
190 SCALE=20:ROT=U:DRAW 2 AT X,Y
200 SCALE=10:XDRAW 2 AT X,Y
210 NEXT U
220 X=174
230 NEXT I
240 REM PABOTA C 3-M ОБРАЗОМ
250 ROT=0:SCALE=5
260 DRAW 3 AT 127,127
270 ROT=8:A=3:B=15:C=2
280 FOR S=A TO B STEP C
290 SCALE=S
300 DRAW 3 AT 127,180
310 FOR T=1 TO 50 : NEXT T
320 XDRAW 3 AT 127,180
330 NEXT S
340 R=A:A=B:B=R:C=-C:GOTO 280
```

Более подробно с вопросами, изложенными в настоящей статье, можно познакомиться в разработке авторов «Факультативные занятия на ПЭВМ «Агат».

Интегрированная среда Си для ПЭВМ «Агат-9»

В последнее время при разработке программ заслуженную популярность приобрел язык Си. В связи с этим, а также учитывая отсутствие компиляторов на «Агате», создание компилятора этого языка является актуальной проблемой. После переработки известного на 8-разрядных машинах Aztec C возникла интегрированная среда компилятора языка Си. При ее разработке мы старались сделать использование компилятора простым как для программистов, работавших ранее на «Агате» и использовавших Бейсик или Ассемблер, так и для привыкших к среде компиляторов Си на IBM PC.

Надеемся, что нам удалось реализовать это, хотя бы частично.

Для работы компилятора необходим «Агат-9» практически любой конфигурации. Один дисковод типа EC-5323.01 — это минимум, а максимум — до 6 дисководов различных типов и принтер. Дополнительная память не используется. Интегрированная среда включает Компилятор, Ассемблер и Линкер. Все вместе они занимают один диск, и то не полностью.

В комплект поставки входят стандартные библиотеки работы с целочисленной и плавающей арифметикой, с устройствами вво-

да/вывода. При желании пользователь может сам создавать собственные библиотеки. В результате работы компилятора появляется исполняемый файл, который может быть запущен на ПЭВМ «Агат 7/9». Естественно, программы на Си будут работать гораздо быстрее, чем на Бейсике, правда несколько медленнее, чем на Ассемблере.

Правильно используя возможности компилятора и библиотек можно добиться высокой скорости выполнения программы, сочетающейся с высокой скоростью ее написания, чему способствует дружественная оконная среда. При необходимости создания некоторых функций, к которым предъ-

являются высокие требования по быстродействию, их можно писать на Ассемблере. Для этого реализована возможность вставки ассемлерных участков в Си-программу.

Линкер позволяет связать любой набор (не более 40 файлов) программ на языке Си и на Ассемблере, содержащих перекрестные ссылки.

Всех, кто заинтересовался этой разработкой, просим обращаться в научно-методический центр «Агат». Наш адрес: 127540, Москва, ул. Дубнинская, д. 23, школа № 231. Телефоны для справок: (095) 489-00-91, 487-86-68.

НПКП «ИНТЕРСЕРВЕР»

разрабатывает и распространяет программные системы для БК-0011М и компьютерных классов, поставляет доступную по цене локальную сеть ВКLink (БК0011М с центральной IBM PC). С момента публикации нашей рекламы в ИНФО № 2—92 появились новые разработки: *сетевая система Sprut3 с поддержкой операционной системы на периферийных компьютерах, оболочка операционной системы Inter-Commander, графический редактор Paint11М, программа печати текстов Kprint, драйвер «мыши», программа копирования файлов с дискет IBM PC и др.*

Адрес: 113162, Москва, М—162, а/я 1.

Телефон: 230-79-78, 522-43-32.



Редактор шрифтов для УКНЦ 97

Редактор шрифтов TYPE предназначен для создания шрифтов, символы которых выводятся на дисплей. Редактор написан на Бейсике для периферийных ЭВМ, объем программы 13 блоков. Редактор порождает загружаемые шрифты, записанные на диск в двоичном формате (с расширением BIN). Такие шрифты можно загрузить из Бейсика командой BLOAD с автозапуском. Имеется возможность преобразования шрифтов из формата .BIN в формат .SAV, что позволяет использовать их при работе с системой RT=11. К редактору прилагается набор разнообразных шрифтов.

В описании редактора используются общепринятые международные обозначения клавиш на клавиатуре.

Таблица соответствия обозначений:

⟨Esc⟩ — ⟨AP2⟩
 ⟨Ctrl⟩ — ⟨УПР⟩

Для запуска редактора надо набрать команду:

LOAD"TYPE",R

Через несколько секунд редактор начнет свою работу: появится заставка. Для продолжения работы — нажать пробел. На экран выводится краткий список

команд редактора. Для дальнейшей работы необходимо нажать пробел. При этом происходит переход в режим выбора.

Режимы редактора:

1. Выбор.

В этом режиме можно выбрать любой символ значогенератора для его редактирования. Выбор производится клавишами управления курсором. Выбрав нужный символ ⟨BK⟩, можно перейти в режим редактирования.

2. Редактор.

В этом режиме можно изменять любой выбранный символ. Он считывается в лупу, расположенную в верхнем левом углу экрана. Каждый символ представлен матрицей 8×11 точек. Переключение в матрице производится с помощью клавиш управления курсором. Поставить точку можно, нажав клавишу ⟨1⟩, стереть точку клавишей ⟨2⟩, инвертировать точку клавишей ⟨3⟩ или клавишей ⟨пробел⟩. Клавиша ⟨4⟩ служит для очистки матрицы символа, а клавиша ⟨5⟩ — для инверсии матрицы. Нажав клавишу ⟨BK⟩, можно запомнить измененный символ и перейти в ре-

жим выбора. Для выхода без записи используйте клавишу ⟨Esc⟩.

3. Запись.

Режим позволяет записать измененный шрифт на диск. Для перехода в режим служит комбинация клавиш ⟨Ctrl⟩+⟨S⟩ (S — начальная буква слова Save — сохранить). На экране появится сообщение: «Имя файла:?».

Теперь надо набрать имя записываемого шрифта, например: «CHAR», и нажать клавишу ⟨BK⟩, шрифт запишется на диск в виде файла с именем CHAR.BIN. После записи происходит автоматический переход в режим выбора.

4. Чтение.

Режим позволяет считать измененный шрифт с диска. Для перехода в режим служит комбинация клавиш ⟨Ctrl⟩+⟨L⟩ (L — начальная буква слова LOAD — загрузить). На экране появится сообщение: «Имя файла:?».

Теперь можно набрать имя считываемого шрифта, например «CHAR», и нажать клавишу ⟨BK⟩. Шрифт считывается с диска, после чего вы автоматически переходите в режим выбора.

5. Помощь.

Получить краткий список команд редактора можно, нажав клавишу <ПОМ>. Выход из него осуществляется нажатием клавиши <пробел>.

Дополнительные возможности:

1. Копирование символа.

Копирование символа производится в режиме выбора. С помощью клавиш управления курсором выбирается исходный символ и нажимается клавиша <C> (C — начальная буква слова Copy — копировать), после чего выбирается символ, на который будет производиться копирование, и нажимается клавиша <BK>.

2. Для удобства работы с редактором в левом нижнем окне печатается название текущего режима работы.

Выход из редактора производится нажатием клавиши <Esc> в режиме выбора. При этом измененный шрифт сохраняется в знакогенераторе.

Использование созданных шрифтов.

Для загрузки шрифта используется команда BLOAD с автозапуском. Например: BLOAD "CHAR", R

Использование шрифтов в программах.

Для автоматизации процесса загрузки мы предлагаем использовать маленькие программы-загрузчики. Например:

```

0 GOSUB 194
1 GOSUB 205
2 COLOR 8,2,2
3 WIDTH 80
4 ? AT(22,5)"+-----+
   +-----+
5 FOR I=6T015
6 ? AT(22,I)"+-----+";AT(56,I)"+-----+
7 NEXT
8 ? AT(22,16)"+-----+
   +-----+
9 ? AT(27,19)"Редактор шрифтов
  для УК-НЦ";AT(32,20)"Сделано
  в ФНЦ";CHR$(80204)+"64"
10 IF I>8THI=4ELI=I+1
11 COLOR I,I,I
12 ? AT(24,7)"+-----+";AT(32,7)
   "+-----+";AT(38,7)"+-----+";AT(40,7)
   "+-----+";AT(48,7)"+-----+";

```

```

AT(24,8)"+-----+";AT(27,8)"+-----+";
AT(30,8)"+-----+";AT(32,8)"+-----+";
13 ? AT(38,8)"+-----+";AT(40,8)"+-----+";
   AT(46,8)"+-----+";AT(48,8)"+-----+";
   AT(54,8)"+-----+";AT(27,9)"+-----+";
   AT(33,9)"+-----+";AT(37,9)"+-----+";
   AT(40,9)"+-----+";AT(46,9)"+-----+";
   AT(48,9)"+-----+";AT(27,10)"+-----+";
   AT(34,10)"+-----+";AT(36,10)"+-----+";
   AT(40,10)"+-----+";
   AT(48,10)"+-----+";
14 ? AT(27,11)"+-----+";AT(35,11)"+-----+";
   AT(40,11)"+-----+";AT(48,11)"+-----+";
   AT(27,12)"+-----+";AT(35,12)"+-----+";
   AT(40,12)"+-----+";AT(48,12)"+-----+";
   AT(27,13)"+-----+";AT(35,13)"+-----+";
   AT(40,13)"+-----+";AT(48,13)"+-----+";
   AT(54,13)"+-----+";
15 ? AT(26,14)"+-----+";AT(34,14)
   "+-----+";AT(40,14)"+-----+";AT(48,14)
   "+-----+";AT(0,0);
16 IF INKEY="" THEN 10 EL
   COLOR 8,2,2
17 GOSUB 209
18 GOSUB 20
19 GOTO 46
20 CLS
21 ? " *** Инструкция ***"
22 ?
23 ? " 1. Режим ВЫБОР"
24 ? "Выбор символа : ";
   CHR$(0);" ";CHR$(1);" ";
   CHR$(2);" ";CHR$(3)
25 ? "Переход в РЕДАКТОР : ";
   CHR$(20)
26 ? "Запись шрифта : Ctrl+S"
27 ? "Чтение шрифта : Ctrl+L"
28 ? "Копирование символа :
   C+выбрать";CHR$(20)
29 ? "Выход : Esc"
30 ?
31 ? " 2. Режим РЕДАКТОР"
32 ? "Выбор точки : ";
   CHR$(0);" ";CHR$(1);" ";
   CHR$(2);" ";CHR$(3)
33 ? "Поставить точку : 1"
34 ? "Стереть точку : 2"
35 ? "Инвертировать точку : 3
   или пробел"
36 ? "Очистить символ : 4"
37 ? "Инвертировать символ : 5"
38 ? "Переход в ВЫБОР : ";
   CHR$(20)
39 ? "Выход без записи : Esc"
40 ?
41 ? " 3. Режимы ЗАПИСЬ
   и ЧТЕНИЕ"
42 ? "Наберите имя файла
   и нажмите : ";CHR$(20)
43 IF INKEY("<>") THEN 43
44 IF INKEY="" THEN 44 EL CLS
45 RETURN
46 ?
47 ? " +-----+
   +-----+
48 ? " | | | |";
   AT(32,2)"+-----+";
49 ? " +-+ | |";
   AT(32,3)"+-----+";
50 FOR I=4T012
51 ? AT(4,I)"+-----+";AT(13,I)"+-----+";
   AT(32,I)"+-----+";
52 NEXT

```

```

53 ? " +-----+ |";
   AT(32,13)"+-----+";
54 ? " +-----+ |";
   AT(32,14)"+-----+";
55 ? " | | |";
   AT(32,15)"+-----+";
56 ? " +-----+
   +-----+";
57 ? AT(6,22)"Для помощи нажмите
   <ПОМ>";AT(12,23)"Выход : Esc"
58 FOR J=2T015
59 FOR I=16T031
60 LOCATE I,J
61 POKE -138,J=16+I-16
62 NEXT I,J
63 X=16
64 Y=2
65 ? AT(5,15)" Выбор "
66 LOCATE X,Y
67 ZH=INKEY
68 IF ZH="" THEN 67 EL Z=ASC(ZH)
69 IF Z=12TH187
70 IF Z=19TH157
71 IF Z=67TH253
72 IF Z=13TH81
73 GOSUB 213
74 IF ZH="" THEN 71
75 IF ZH="" THEN 37
76 IF X=15THX=31
77 IF X=32THX=16
78 IF Y=1THY=15
79 IF Y=16THY=2
80 GOTO 66
81 W=Y+16+X-16
82 YY=Y
83 XX=X
84 IF F=1TH256
85 LOCATE 3,2
86 POKE -138,W
87 GOSUB 221
88 FOR J=1T011
89 FOR I=1T08
90 LOCATE I+4,J+1
91 POKE -138,46+A(I,J)*81
92 NEXT I,J
93 X=5
94 Y=2
95 ? AT(5,15)"Редактор"
96 LOCATE X,Y
97 ZH=INKEY
98 IF ZH="" THEN 97
99 IF ZH=CHR$(13) THEN 140
100 IF ZH="" THEN 110
101 IF ZH="" THEN 128
102 IF ZH="" THEN 3"ORZH="" THEN 136
103 IF ZH="" THEN 4"TH235
104 IF ZH="" THEN 5"TH244
105 GOSUB 213
106 IF ZH="" THEN 17
107 IF ZH="" THEN 113
108 IF X=4THX=12
109 IF X=13THX=5
110 IF Y=1THY=12
111 IF Y=13THY=2
112 GOTO 96
113 LOCATE 3,2
114 POKE -138,W
115 X=XX
116 Y=YY
117 LOCATE X,Y
118 POKE -138,W
119 GOTO 65
120 A(X-4,Y-1)=1
121 POKE -608,-31165+80*Y

```

```

122 Q=PEEK(-606)
123 Q=Q OR 2^(X-5)
124 IF X<12THQ=Q OR 2^(X+3)EL
    Q=Q OR-32768
125 POKE -606,Q
126 POKE -138,127
127 GOTO 96
128 A(X-4,Y-1)=0
129 POKE -608,-31165+80*Y
130 Q=PEEK(-606)
131 Q=Q AND NOT(2^(X-5))
132 IF X<12THQ=Q AND
    NOT(2^(X+3))EL
    Q=Q AND NOT-32768
133 POKE -606,Q
134 POKE -138,46
135 GOTO 96
136 IF A(X-4,Y-1)=0TH120 EL128
137 GOSUB 205
138 CLS
139 END
140 LOCATE 3,2
141 POKE 120,128
142 POKE 144,6400+W*12
143 POKE 132,6242+2*W
144 POKE 134,144
145 POKE 136,1
146 POKE 98,120
147 Z=USR1(0)
148 FOR I=1TO11ST2
149 Q=A(1,I)+A(2,I)*2+A(3,I)*4+
    A(4,I)*8+A(5,I)*16+A(6,I)*32+
    A(7,I)*64+A(8,I)*128+A(1,I+1)*
    *256+A(2,I+1)*512+A(3,I+1)*
    *1024+A(4,I+1)*2048+A(5,I+1)*
    *4096+A(6,I+1)*8192+A(7,I+1)*
    *16384+A(8,I+1)*32768
150 IF Q>32767THQ=Q-65536
151 POKE 143+I,Q
152 NEXT
153 POKE 132,6400+W*12
154 POKE 136,6
155 Z=USR1(0)
156 GOTO 113
157 ? AT(5,15)" Запись "
158 GOSUB 231
159 A=-16384
160 DATA 5568,0,-29729,-500,
    -32515,-23096,254,771,-27617,
    -578,503,135,0,-1,254,4096,26
    ,6306
161 RESTORE 160
162 FOR I=0TO17
163 READ K
164 POKE A+I*2,K
165 NEXT
166 POKE A+2,A+24
167 POKE A+24,A+30
168 POKE A+36,A+50
169 POKE A+38,1610
170 DATA -29417,224,5568,6306,
    5599,24601,-504,5569,1600,
    5151,-500,2719,-504,32325,
    -29417,0,135
171 FOR I=0TO16
172 READ K
173 POKE 144+I*2,K
174 NEXT
175 POKE 98,120
176 POKE 120,128
177 POKE 132,10240
178 POKE 134,144
179 POKE 136,17
180 Z=USR1(0)

```

```

181 POKE 120,138
182 POKE 142,10240
183 Z=USR1(0)
184 IF PEEK(-580)<>128TH184
185 BSAVE FH,A,A+3224
186 GOTO 63
187 ? AT(5,15)" Чтение "
188 GOSUB 231
189 BLOAD FH
190 DEF USR2=-16384
191 Z=USR2(0)
192 IF PEEK(-580)<>128TH192
    EL50
193 GOTO 58
194 DIM A(8,12)
195 DATA 0,6111,80,-608,6111,
    -606,80,135,5568,0,-29729,
    -580,-32515,-24041,254,771,
    -27617,-578,503,135,0,-1,
    254,0,4096,26,0,0,0,6144,26
196 RESTORE 195
197 FOR I=0TO30
198 READ K
199 POKE 80+I*2,K
200 NEXT
201 BEEP
202 DEF USR=82
203 DEF USR1=96
204 RETURN
205 POKE -138,27
206 POKE -138,166
207 POKE -138,49
208 RETURN
209 POKE -138,27
210 POKE -138,166
211 POKE -138,50
212 RETURN
213 IF ZH<>CHR(27)TH220
214 ZH=INKEYH
215 IF ZH=""TH220
216 IF ZH="A"THY=Y-1
217 IF ZH="B"THY=Y+1
218 IF ZH="C"THX=X+1
219 IF ZH="D"THX=X-1
220 RETURN
221 FOR J=1TO11
222 POKE 80,-31085+J*80
223 Z=USR(0)
224 FOR I=1TO8
225 Q=PEEK(80)
226 IF Q<0THQ=Q+65536
227 Q=INT(Q/256)/2^I
228 A(I,J)=INT((Q-INT(Q))*2)
229 NEXT I,J
230 RETURN
231 ? AT(16,19)"Имя файла:"+
    CHR(11):
232 INPUT FH
233 IF FH=""ORLEN(FH)>6TH231
234 RETURN
235 FOR I=1TO8
236 FOR J=1TO12
237 A(I,J)=0
238 NEXT J,I
239 FOR I=0TO10
240 POKE -608,-31085+I*80
241 POKE -606,0
242 NEXT
243 GOTO 88
244 FOR I=1TO8
245 FOR J=1TO12
246 A(I,J)=(A(I,J)+1)MOD 2
247 NEXT J,I
248 FOR I=0TO10

```

```

249 POKE -608,-31085+80*I
250 POKE -606,NOT PEEK(-606)
251 NEXT
252 GOTO 88
253 Q=Y*16+X-16
254 F=1
255 GOTO 66
256 F=0
257 LOCATE 3,2
258 POKE -138,Q
259 GOSUB 221
260 GOTO 140

```

Преобразование шрифтов.
Для использования шрифтов в системе необходимо преобразовать их из типа .BIN в тип .SAV. Для этой цели используется программа TYPE.SAV.

```

10 BLOAD "CHAR",R
    ' загрузка шрифта
20 LOAD "PROGR",R ' загрузка
    программы использующей шрифт

```

```

1 WIDTH 64
2 ? "Преобразователь шрифтов
    из типа .BIN в тип .SAV"
3 INPUT "Входной файл ";NH
4 INPUT "Выходной файл ";MH
5 ? "Преобразование ..."
6 BLOAD NH
7 FOR IZ=&0137000
    TO &0137776 ST2
8 POKE IZ,0
9 NEXT
10 POKE &0137040,&01000
11 POKE &0137042,&01000
12 POKE &0137050,&07300
13 POKE &0137360,-1
14 POKE &0137362,&0360
15 POKE &0140002,&01030
16 POKE &0140030,&01036
17 POKE &0140044,&01062
18 POKE &0140026,&0104350
19 BSAVE MH,&0137004,&0146300
20 ?
21 ? "Преобразование окончено"

```

Для загрузки этой программы наберите команду LOAD "TYPE.SAV", R. Через несколько секунд появится сообщение:

«Преобразователь шрифтов из типа .BIN в тип .SAV «Имя шрифта:?»»

Вы должны набрать имя шрифта, который подлежит преобразованию, например "CHAR", и нажать клавишу <BK>. Появится сообщение: «Имя шрифта после преобразования:?»»

Вы должны набрать имя шрифта после записи, например "CHAR1", и нажать кла-

вишу (BK). Появятся сообщения:

«Преобразование...»

«Преобразование завершено».

«Еще раз (Y/N)?»

К сожалению, Бейсик позволяет записать шрифт только в типе .BIN. Для использования шрифта в системе необходимо переименовать файл командой `RENAME CHAR1.BIN CHAR.SAV`. Теперь, запуская программу `CHAR.SAV` в системе, вы можете загрузить шрифт.

Список разработанных нами шрифтов:

`INIT.BIN` — стандартный шрифт УКНЦ (преобразовывать в тип .SAV нельзя);

`CGA.BIN` — шрифт CGA

компьютера `IBM PC` (маленький жирный);

`BETA.BIN` — шрифт компьютера `IBM PC` (большой жирный);

`THIN.BIN` — шрифт компьютера `IBM PC` (большой тонкий);

`ORN.BIN` — (маленький жирный угловатый);

`DOUBLE.BIN` — двойной стандартный шрифт УКНЦ;

`GREK.BIN` — греческий (маленький тонкий).

С. КЛОКОВ, Д. ЛАНИН
ФМШ № 64, г. Омск

Укажи его свойство: ловит мышей

Укажи его свойство: нет
Затем экран очищался и диалог продолжался в такой форме:

4 лапы? Да

Хвост? Нет

ТАКОГО ЖИВОТНОГО Я НЕ ЗНАЮ! КТО ЭТО!

НАЗОВИ ЖИВОТНОЕ:

Если все ответы были утвердительными, то на экране высвечивался текст: **«ВАШЕ ЖИВОТНОЕ: Кот!»**

Программа была продемонстрирована участникам конференции учителей. При этом обнаружили некоторые неудобства в пользовании. Например, если задуманное животное (после ранее введенного «Кот») было «Кошка», то, проверив несколько первых свойств, характерных как для кота, так и для кошки, машина выдавала: **«ВАШЕ ЖИВОТНОЕ: Кот!»** Чтобы изменить порядок задания вопросов или ввести дополнительные свойства, требовалось начинать работу заново. Поэтому для обеспечения удобства работы с данными потребовалось обновление системы.

Продолжая совершенствовать программу, мы постепенно расширяли ее возможности. Появились новые режимы работы. В результате эксперимента убедились, что игра «Зоопарк» может послужить основой для создания моделей экспертных систем по различным общеобразовательным дисциплинам в школе. Так появилась идея создания программы «Систематизация».

Она заключается в следующем. Пользователю предлагаются пять режимов работы, которые выбираются с помощью «Фиксирующей рамки»: диалог, редактор, терминал, помощь, поиск.

Режим «Диалог» основной. В нем создается база данных по принципу игры «Зоопарк». В этом режиме используется следующая

Программа «Систематизация» — модель экспертной системы

В статье академика Е. П. Велихова «Новая информационная технология в школе» (ИНФО. 1986. № 1) говорится о необходимости широкого использования компьютера как учебного средства.

В качестве одного из перспективных направлений предлагается идея, заключающаяся в том, что учащийся выполняет функцию учителя, закладываящего знания в машину» [1, с. 22]. Примером ее реализации является «Игра в животных», суть которой заключается в том, что с помощью свойств, характеризующих данное животное, и элементарных ответов ученика на вопросы, формулируемых на основе введенных в компьютер свойств в форме «да — нет», последний пытается установить задуманное учеником животное.

Идею этой игры мы попытались реализовать в про-

грамме под названием «Зоопарк». Начиналась игра так. Компьютер давал задание «Назови животное:». После получения текста (названия животного) предлагал указать его свойства. Затем компьютер начинал проверять свои знания, задавая играющему характеристические вопросы, формулируемые по текстам введенных свойств. Если в это время ученик задумывал другое животное, которое машина не знала, и на один из предложенных вопросов отвечал «Нет», на экране высвечивалось: **«ТАКОГО ЖИВОТНОГО Я НЕ ЗНАЮ! КТО ЭТО!»** — и игра продолжалась дальше.

Например, при первоначальном запуске программы возник следующий диалог:

НАЗОВИ ЖИВОТНОЕ: Кот

Укажи его свойство: 4 лапы

Укажи его свойство: хвост

Укажи его свойство: усы

Укажи его свойство: любит

молоко

структура:

Назови понятие:

Укажи его свойство:

Ваше понятие:
или

В моей базе данных такого понятия нет!

Режим «Редактор». Позволяет просмотреть имеющиеся в базе данных понятия и соответствующие им свойства. При желании пользователь в этом режиме может убрать свойство, изменить его или добавить новое.

Режим «Терминал». Позволяет записать созданную базу данных на диск или считать с диска.

Режим «Помощь». Содержит образец заполнения базы данных, описаны принцип работы с программой и возможности каждого режима.

Режим «Поиск». Позволяет оперативно находить все понятия, обладающие запрошенным свойством, или свойства, относящиеся к запрошенному понятию.

Экран имеет вид, показанный на рис. 1.

1

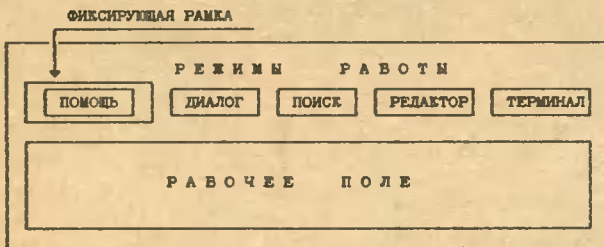


Рис. 1

Программу можно использовать по различным общеобразовательным дисциплинам на таких этапах учебного процесса, как закрепление, обобщающее повторение, контроль, самостоятельная работа учащихся.

Например, в VIII классе по теме «Параллелограмм и его свойства» сильные ученики могут самостоятельно заполнить базу данных по свойствам параллелограмма на этапе изучения нового материала, после чего учитель проверяет их работу и записывает на диск созданную базу данных для дальнейшего использования при закрепле-

нии, повторении и для работы со слабыми учениками. Аналогично при завершении изучения главы «Многоугольники» на диске может накапливаться материал, позволяющий контролировать

собие при подготовке к контрольным работам и экзаменам. На рис. 2 показана схема (бинарное дерево), позволяющая реализовать в диалоге соотношение понятий и их свойств в данной системе.

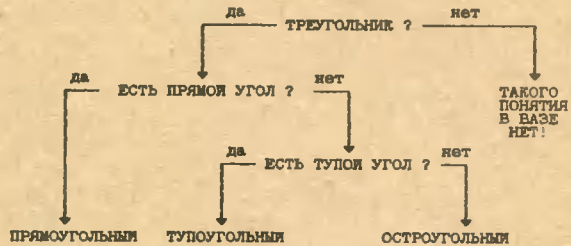


Рис. 2

ляет достаточно эффективно влиять на систематизацию знаний у школьников.

Разработка выполнена студентами IV курса математического факультета Омского государственного педагогического института им. А. М. Горького под руководством ст. преподавателя В. Буцик.

В настоящее время данная система реализована на УМКЦ, УАМАНА, КУВТ-86 (язык BASIC). Разрабатывается вариант системы на IBM (язык FoxBASE).

Литература

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

**В. БУЦИК, Е. КУНСТ,
Е. БУДЕЛЕВА, Н. ТАРАНЕНКО**

2

Программа «Спектр»

Программа «Спектр» была представлена на проходившем в 1991 г. в Красноярске Всесоюзном конкурсе «Юный программист» и признана лучшей среди игровых. Жюри отметило яркие краски и динамизм игры.

Основная цель игровой программы «Спектр» — развитие навыков быстрого принятия решений в постоянно меняющейся обстановке на игровом поле. Игра по сути близка к задачам-головоломкам и помогает играющему развивать образное мышление и тренировать зритель-

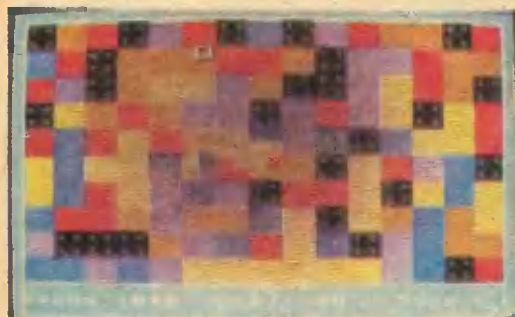
ний. При этом не следует наступать на «запретный» цвет, помеченный точками. Цвет фигуры становится «запретным» при переходе игрока на другую фигуру, при этом цвет, бывший «запретным» ранее, восстанавливается.

Таким образом в игре реализована идея динамически меняющегося лабиринта. В игре предусмотрены дополнительные помехи: случайное перекрашивание клеток и сдвиг строк поля. При прохождении этапа экономленное время пересчитыва-

ется в очки. видеопроцессора MSX возможностей: изменение палитры цвета (RGB), шаблонов символов и цвета символов в текстовом режиме, а также спрайты.

При создании программы использовался графический пакет SPRITE. PAS. Первоначальный вариант пакета был написан в МГУ им. М. В. Ломоносова, в процессе работы в ШЮП г. Протвино он был расширен и модифицирован.

Программа написана в соответствии с принципами структурного программирования, т. е. разбита на до-



ную память. Игровым полем является экран монитора, заполненный разноцветными фигурами, составленными из клеток одного цвета. Игра состоит из 12 этапов, и условием ее связано с уменьшением числа используемых цветов с 6 до 4 и размеров фигур. Особый интерес программа «Спектр» вызывает у ребят 11—13 лет.

Задача играющего — за определенное время пересечь игровое поле из левого верхнего угла в правый ниж-

ется в очки.

Игра заканчивается при выполнении одного из трех условий: игрок попал на «запретный» цвет, кончилось время, отведенное на прохождение этапа, игрок прошел все двенадцать этапов. На время работы программа поддерживает таблицу рекордов по очкам.

Программа «Спектр» написана для ЭВМ MSX-1 на языке высокого уровня TURBO PASCAL. Она использует несколько специфических для

статочны независимые блоки и не содержит ни одного оператора GOTO, так как логические структуры IF/THEN/ELSE, WHILE/DO, REPEAT/UNTIL полностью обеспечивают потребности алгоритма и делают его более наглядным.

С. БИТЮКОВ,
X кл., СШ № 2,
г. Протвино

В. КИРЮХИН

IV Всероссийская олимпиада школьников по информатике

Заключительный этап Всероссийской олимпиады школьников по информатике проводился в этом году с 22 по 27 марта в г. Троицке Московской области на базе центра информатики «Байтик». В олимпиаде приняли участие 102 школьника из более чем 70 территорий Российской Федерации, включая Москву, Санкт-Петербург и Троицк. Возглавлял оргкомитет олимпиады председатель комитета по народному образованию администрации Московской области В. М. Егоров. Председателем жюри являлся заместитель директора Троицкого института инновационных и термоядерных исследований Д. Н. Соболенко.

Благодаря усилиям организаторов и особенно центра информатики «Байтик», возглавляемого его директором В. Е. Дудочкиным, нынешняя олимпиада соответствовала всем требованиям, предъявляемым к национальным олимпиадам. На обоих турах всем участникам предоставлялись однотипные персональные компьютеры IBM PC XT. В качестве систем программирования допускалось использование Turbo Pascal 5.0, Turbo Pascal 5.5, Quick Basic, GW-Basic, Microsoft C, Quick C, Turbo C++.

Существенно отличался в этом году порядок проведения олимпиады. Как и на всех прошлых международных и всесоюзных олимпиадах по информатике, оба тура здесь были равноправными. На каждом туре участникам предлагалось по одной задаче и для ее решения отводилось 4 ч. Во время туров разрешалось пользоваться любой литературой. Запрещалось только использовать дискеты с нестандартным программным обеспечением.

На рассмотрение жюри на этой олимпиаде были представлены восемь задач, каждая из которых имела многоуровневый характер, т. е. содержала ряд подзадач с различной степенью сложности, объединенных общей идеей. Такой подход к формированию олимпиадных задач позволил участникам более четко сосредоточиться на раскрытии основной идеи задачи и продемонстрировать свои способности на всех этапах решения поставленной проблемы, включая формализацию, алгоритмизацию и програм-

мирование.

После тщательного обсуждения предложенных на рассмотрение задач членами жюри были отобраны следующие.

Задача I тура ЛОМКА (автор В. В. Прохоров)

Пусть W — множество всех замкнутых односвязных ломаных на координатной плоскости xOy , удовлетворяющих условиям:

а) ломаная ломается только под прямым углом;

б) звено ломаной параллельно одной из осей координат;

в) отсутствуют самопересечения или самокасания ломаной;

г) $(x_1, y_1), \dots, (x_N, y_N)$ — целочисленные координаты всех точек, где ломаная претерпевает излом (порядок нумерации произволен и неизвестен), N — количество изломов.

Требуется:

А. Написать по возможности оптимальные (по времени исполнения) программы (с обоснованием алгоритмов), которые по задаваемым N и $(x_1, y_1), \dots, (x_N, y_N)$ выдавали бы и отображали на экране монитора:

1) какую-либо ломаную из множества W ;

2) ломаную из множества W , имеющую наибольшую длину, и значение этой длины;

3) ломаную из множества W , ограничивающую наибольшую площадь, и значение этой площади;

4) количество ломаных в множестве W .

Б. Решить задачу А, исключив пункт б из определения W . **Примечание:** односвязной называется ломаная, из любой точки которой можно попасть в любую другую ее точку, двигаясь по этой ломаной.

Подробный разбор задачи II тура дается в статье Е. Андреевой и А. Марченко «Олимпиадная задача» (см. с. 105).

Обычно на всех олимпиадах по информатике предлагаемые задачи вызывают повышенный интерес со стороны участников олимпиады и их руководителей. Одни связывают свои невысокие результаты с неудачным выбором задач. Другие сетуют на то, что задачи носили программистский

характер и нельзя от школьников требовать получения работоспособной программы: достаточно представить только описание алгоритма. Есть и такие, которые считают наоборот.

Конечно, хорошо, когда олимпиадная задача удовлетворяет всем запросам. Однако от участников требуется проявить свои умения и навыки именно при решении представленной задачи. И в этом случае все оказываются в одинаковых условиях. Вопрос, нравится или не нравится задача, должен отходить во время олимпиады на второй план.

На прошедшей олимпиаде претензий к задачам практически не было. Общее мнение — предложенные задачи полностью удовлетворяли всем требованиям, характерным для олимпиад такого уровня. И даже тот факт, что многие участники выступали не так, как хотелось бы, большинство руководителей команд связывали не со сложностью задач, а с недостаточным уровнем подготовки и низким уровнем проведения олимпиад в ряде областей и территорий Российской Федерации.

При проверке каждая задача оценивалась исходя из 100 баллов. Участник мог на двух турах набрать максимально 200 баллов. Оценка задачи I тура осуществлялась с помощью тестов и путем анализа представленных описаний с обоснованиями наилучших алгоритмических решений, как это требовалось в условии. Наилучшие результаты при решении этой задачи показали:

Давыдок Дмитрий (Санкт-Петербург) — 87 баллов;

Иоффе Сергей (Московская обл.) — 65 баллов;

Офенбах Игорь (Омск) — 59 баллов.

Задача II тура оценивалась из предположения, что правильность представленных алгоритмов должна подтверждаться результатами работы программы. С этой задачей многие участники олимпиады справились лучше, чем с задачей предыдущего тура. Наилучший результат здесь показали:

Кузнецов Евгений (Москва) — 61 балл;

Ласкин Андрей (г. Фрязино, Московская обл.) — 54 балла;

Шарифуллин Марат (г. Усинск, Коми) — 53 балла.

Победителями олимпиады стали:

Иоффе Сергей (п. Чернооголовка, Московская обл.) — 117 баллов;

Давыдок Дмитрий (Санкт-Петербург) — 113 баллов;

Елизаров Роман (Санкт-Петербург) — 104 балла.

Среди учащихся XI класса 5 человек получили диплом I степени (они набрали 90—117 баллов), 6 человек — диплом II степени (63—84 балла), 6 человек — диплом III степени (48—61 балл). Диплом III степени получила также единственная на олимпиаде девушка — Никитина Елена (г. Екатеринбург). Она же была награждена специальным призом центра информатики «Байтик».

Среди учащихся невыпускных классов диплом I степени получили 2 участника (91—104 балла), диплом II степени — 4 (55—67 баллов) и диплом III степени — 5 (31—50 баллов). Наряду с дипломами многие победители олимпиады получили памятные призы и подарки.

Победители IV Всероссийской олимпиады по информатике

Среди учащихся XI класса получили

дипломы I степени:

Иоффе Сергей (п. Чернооголовка Московской обл., СШ № 82);

Давыдок Дмитрий (Санкт-Петербург, СШ № 239);

Кузнецов Евгений (Москва, СУНЦ МГУ);

Жуков Дмитрий (Москва, СУНЦ МГУ); Россиев Алексей (Красноярск, СШ № 10).

Дипломы II степени:

Гузев Алексей (г. Пермь, ФМШ);

Офенбах Игорь (г. Омск, СШ № 64);

Тучинский Сергей (г. Воронеж, СШ № 15);

Гладышев Павел (г. Челябинск, СШ № 31);

Шарифуллин Марат (г. Усинск, Коми, СШ № 4);

Черенков Алексей (г. Нальчик, лицей).

Дипломы III степени:

Пер Юрий (г. Тула, СШ № 73);

Ласкин Андрей (г. Фрязино Московской обл., СШ № 1);

Шестаков Александр (г. Оренбург, гимназия);

Кунков Сергей (г. Екатеринбург, лицей);

Хохолов Владислав (г. Сургут, Тюменская обл.);

Меньков Алексей (г. Костомукша, СШ № 2);

Никитина Елена (г. Екатеринбург, лицей).

Среди учащихся VIII—X классов получили

дипломы I степени:

Елизаров Роман (Санкт-Петербург, СШ № 30);

Шевяков Родион (г. Волгоград, СШ № 78).

Дипломы II степени:

Кленин Александр (г. Владивосток, СШ № 23);

Миронов Илья (Санкт-Петербург, СШ № 239);

Виноградов Андрей (г. Рыбинск, СШ № 1);

Бычков Сергей (Москва, СУНЦ при МГУ).

Дипломы III степени:

Нестеренко Олег (г. Нальчик, СШ № 4);

Черняховский Андрей (г. Нижний Новгород, СШ № 55);

Орлов Леонид (г. Чебоксары, СШ № 27);

Каляткин Игорь (г. Бийск, Алтайский край, ДТТ);

Каменский Никита (г. Бийск, Алтайский край, ДТТ).

Олимпиадная задача

Рассмотрим задачу (авторы Е. Андреева и А. Марченко), предлагавшуюся во II туре IV Всероссийской олимпиады по информатике.

Компостер

В билете пассажира оказалось пробито отверстий больше, чем штырей в компостере. Пассажир утверждал, что пользовался только одним компостером, но случайно нажал на него несколько раз. Контролеру требуется определить, могло ли быть получено заданное расположение отверстий одним и тем же компостером, если билет можно пробивать с обеих сторон неограниченное число раз и произвольно перемещать и поворачивать относительно компостера. Пробитые отверстия не выходят за пределы билета. В билете было пробито N ($N \leq 9$) отверстий.

Требуется:

А. Для компостера с двумя штырями ($S=2$) составить программу, которая:

1. Определяет, можно ли получить заданным компостером требуемое расположение отверстий в билете. Если это возможно, то изображает вид билета после каждого нажатия компостера. В противном случае выводит соответствующее сообщение.

2. Определяет количество K различных компостеров, каждым из которых можно пробить заданную конфигурацию.

3. При $K=0$ (см. п. 2) находит компостер, с помощью которого можно пробить наибольшее количество из заданных отверстий.

4. Находит минимальное число нажатий, требуемое для пробивки заданной конфигурации отверстий, для каждого компостера из п. 2.

Б. Решить задачу А для компостеров с числом штырей S ($S > 2$).

Примечания.

Все исходные данные — натуральные числа.

Компостеры, дающие при однократном нажатии совпадающие конфигурации отверстий, считаются одинаковыми.

Относительное расположение отверстий в билете и штырей в компостере вводится либо с клавиатуры, либо из файла с именем COMP.DAT.

Структура вводимой информации: $\{N, x_1, y_1, \dots, x_N, y_N, S, u_1, v_1, \dots, u_S, v_S\}$, где x_p, y_p — координаты отверстий в билете, u_i, v_i — координаты штырей в компостере.

Нажатие компостера (см. п. 1) моделировать клавишей «Пробел».

При выводе конфигурации на экран (см. п. 1,3) изображать координатную сетку. При этом программа должна осуществлять подходящее масштабирование.

Идея решения задачи

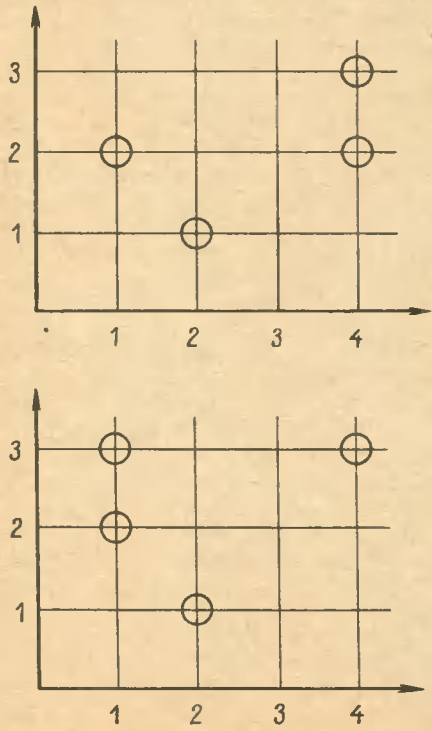
Рассмотрим решение данной задачи для компостера с двумя штырями ($S=2$). Конфигурацию из N отверстий в билете будем обозначать при помощи квадратной матрицы B порядка N , каждый элемент $B(i, j)$ которой равен квадрату расстояния между центрами i -го и j -го отверстий, т. е. $B(ij) = (x_i - x_j)^2 + (y_i - y_j)^2$. Матрица B симметрична относительно главной диагонали. Использование квадратов расстояний, а не расстояний между отверстиями позволяет избавиться от погрешностей, связанных с машинным вычислением квадратного корня. Любой компостер с двумя штырями можно однозначно определить с помощью квадрата расстояния r^2 между его штырями. Тогда для ответа на вопрос п. 1 задачи А необходимо и достаточно проверить, содержится ли число r^2 , соответствующее заданному компостеру, в каждой строке матрицы B .

В п. 2 величины возможных расстояний между штырями компостеров заранее неизвестны, но вследствие того, что компостер, которым можно пробить все отверстия, обязательно пробивает и первое, достаточно рассмотреть числа только из первой строки матрицы B . Таким образом, количество K различных компостеров равно количеству различных чисел из первой строки матрицы B , каждое из которых, аналогично п. 1, присутствует во всех остальных строках матрицы квадратов расстояний. Очевидно, что $K < N$.

Если не существует компостера, которым можно пробить все N отверстий ($K=0$), то приступаем к поиску максимального числа отверстий N_{\max} , которые можно пробить одним компостером. Всего существует не более $N(N-1)/2$ различных расстояний между отверстиями и соответственно различных компостеров. Искомым компостером будет тот, для которого r^2 встретится в максимальном количестве строк матрицы B , а количество таких строк равно N_{\max} .

При решении п. 4 для каждого найденного ранее в п. 2 компостера надо найти всевозможные m пар отверстий, расстояние между которыми равно расстоянию между штырями этого компостера. Достаточно

перебрать $2^m - 1$ вариантов комбинаций нажатий, причем единица в двоичном представлении номера варианта, стоящая, например, в третьем разряде, означает, что в данном варианте третья пара отверстий должна быть проколота. Перебор вариантов комбинаций нажатий можно сократить, если учесть, что, во-первых, минимальное число нажатий, необходимое для пробития всех отверстий, заключено в интервале от $\lceil (N+1)/2 \rceil$ до $N-1$, т. е. не рассматривать варианты, в номерах которых число единиц двоичного представления меньше $\lceil (N+1)/2 \rceil$, во-вторых, если в какой-то строке матрицы B число r^2 встречается только один раз, то необходимо обязательно пробить эту пару отверстий и количество вариантов нажатий уменьшится в два раза. Итак, подходящим будет вариант с минимально возможным числом единиц в двоичном представлении номера варианта. Перейдем к решению задачи для компостеров с числом штырей более двух. Первая проблема заключается в том, как определить, какие конфигурации являются одинаковыми. Для компостеров с двумя штырями два компостера одинаковы, если равны расстояния между их штырями. В случае $S=3$ взаимное расположение штырей однозначно определяется набором трех расстояний между тремя парами штырей. Но уже при $S=4$ это не так. На рис. пока-



заны две различные конфигурации с одинаковым набором расстояний между штырями в каждой из них. Данный пример взят из теста 9.

Аналогично конфигурации отверстий каждой конфигурацию штырей компостера можно описать с помощью квадратной матрицы квадратов расстояний между штырями порядка S . Очевидно, что один и тот же компостер описывается различными матрицами в зависимости от порядка нумерации штырей. Но каждой строке одной из таких матриц соответствует строка другой матрицы, т. е. между элементами этих двух строк существует взаимно однозначное соответствие. Итак, два множества по S точек на плоскости имеют одинаковую конфигурацию, если существует взаимно однозначное соответствие (биекция) между наборами $S-1$ расстояний от каждой точки до остальных в одном множестве и наборами $S-1$ расстояний между точками в другом множестве.

Другой путь для определения эквивалентности конфигураций заключается в сравнении множества всех треугольников (всех троек расстояний между точками) из одной конфигурации с множеством треугольников в другой конфигурации. Этих треугольников не более числа сочетаний из S элементов по три, т. е. не более 84. Конфигурации одинаковы, если существует взаимно однозначное соответствие между множеством треугольников в них (понятие эквивалентности двух треугольников введено выше).

Чтобы сравнить конфигурации между собой и отобрать среди них одинаковые, сначала надо перебрать все возможные конфигурации. Их общее число равно числу сочетаний из N элементов по S , а так как по условию задачи $N \leq 9$, то всего существует не более 126 конфигураций. Пусть среди них мы нашли m одинаковых. Пронумеруем их от 1 до m . По условию задачи все одинаковые конфигурации пробиваются одним компостером, и, следовательно, за одно нажатие этим компостером пробиваются отверстия с номерами, соответствующими одной из m рассматриваемых конфигураций.

Отверстия можно пробивать, пробивая только первую из одинаковых конфигураций (1 нажатие), только вторую (1 нажатие), совместно первую и вторую (два нажатия) и т. д. Всего существует $2^m - 1$ вариантов комбинаций нажатий. Необходимо для каждого из этих вариантов подсчитать количество нажатий и число пробитых при этом отверстий. Количество нажатий равно числу единиц в двоичном представлении номера варианта нажатий. Далее

уже нетрудно отобрать комбинации, для которых число пробитых отверстий равно N , и выбрать из них комбинацию с минимальным числом нажатий. Если таких комбинаций не нашлось, то необходимо запомнить, какое максимальное число отверстий N_{\max} может пробить данный компостер. После этого перейдем к анализу следующей группы одинаковых конфигураций, если такая имеется. Изложенный подход годен и для случая $S=2$.

Программная реализация алгоритма

Для программной реализации мы выбрали алгоритмический язык Turbo Basic, чтобы показать, что и на Бейсике можно написать компактную и достаточно эффективную программу. В обобщенном виде программа, реализующая вышеизложенную идею, имеет простую структуру и представляет собой последовательность выполнения следующих действий:

- 1) чтения данных из файла данных и контроля входных данных;
- 2) генерации всех конфигураций (сочетаний);
- 3) нахождения одинаковых конфигураций;
- 4) перебора всех вариантов комбинаций нажатий с нахождением компостеров, которыми можно пробить все отверстия, их количества, минимального числа нажатий и, если таких компостеров нет, нахождением компостеров с максимальным числом пробитых отверстий (*nmaxgloб*).

В программе отсутствует проверка заданного компостера, так как задание п. 1 является частным случаем общей задачи. Читателю нетрудно самому разобраться в работе программы и промоделировать вывод на экран последовательности пробитий заданным компостером.

В программе использовались следующие двумерные массивы: с (126, 10) — массив всевозможных конфигураций, е (126, 20) — массив, каждая строка которого содержит одинаковые конфигурации, и г (9, 9) — массив квадратов расстояний. Развернутость массивов выбрана так, потому что всевозможных конфигураций не более 126, а отличающихся друг от друга (неодинаковых) еще меньше. Например, в тестах их число не превышает 23. Число элементов строки в массиве е (126, 20) не превышает 16 даже в наихудшем случае, когда отверстия расположены в виде квадрата 3×3 . Поэтому 20-й столбец можно использовать для запоминания числа одинаковых конфигураций, т. е. числа ненулевых элементов в этой строке.

Оптимизация программы заключается в первоочередном выполнении очевидных действий и уменьшении множества вариантов, среди которых ведется перебор. Например, если при поиске минимального числа нажатий P_{\min} после отметки всех элементов матрицы расстояний, которые входят в матрицу конфигурации компостера, оказа-

Таблица 1

	Тесты	Можно К	P_{\min}	V_{\max}
1	N=1 1,1 S=2 1,1 1,3	Нет	—	—
2	N=4 1,1 1,2 1,3 1,4 S=2 1,1 2,1	Да	2	—
3	N=5 5,1 3,3 11,5 1,5 11,9 S=2, 1,1 1,11	Да	1	3
4	N=5, 1,1 2,2 5,1 5,3 4,4 S=2 1,1 2,4	Нет	0	4
5	N=6 1,1 5,1 3,2 1,5 5,5 3,6 S=2 1,1 1,3	Нет	2	3 4
6	N=6 1,1 1,2 1,3 1,4 1,5 1,6 S=2 1,1 5,1	Нет	3	3 4 3
7	N=6 1,1 3,1 2,2 3,3 1,3 1,6 S=2 1,1 2,2	Нет	0	5
8	N=5 1,1 6,1 7,3 5,4 3,5 S=3 1,1 5,4 3,5	Да	1	2
9	N=6 2,1 4,2 1,2 4,3 1,3 3,4 S=4 2,1 1,2 1,3 2,4	Нет	6	2
10	N=7 1,1 1,2 1,4 2,2 2,3 3,1 3,3 S=4 1,1 1,2 2,1 2,2	Нет	3	2 2 3
11	N=6 1,1 3,1 2,2 3,3 1,3 1,6 S=3 1,1 2,1 1,2	Нет	0	5
12	N=6 1,1 3,1 2,2 3,3 1,3 1,6 S=4 1,1 2,2 1,3 1,2	Нет	0	5

лось, что в какой-либо i -й строке матрицы расстояний находятся ровно $S-1$ отмеченных элементов, то существует единственный вариант пробития i -го отверстия и вариантов нажатий становится в два раза меньше ($2^{m-1}-1$). Далее из матрицы исключаются строки и столбцы с номерами отверстий, входящих в найденное нажатие. Процесс исключения строк, может быть, можно продолжить и значительно уменьшить число вариантов комбинаций нажатий. Или рассматривать только те варианты нажатий, при которых количество нажатий (количество единиц в двоичном представлении варианта нажатий) не меньше, чем $[(N+S-1)/S]$, и не больше, чем $N-S+1$. Или при нахождении максимального числа пробитых отверстий $n_{maxglob}$ можно использовать информацию об уже найденных n_{max} , полученную при анализе предыдущих вариантов нажатия.

В прилагаемой программе не реализованы предложения по оптимизации, так как время выполнения тестов на IBM PC не превышает 30 с. Для проверки программы использовались тесты, которые вместе с результатами их выполнения приведены в табл. 1.

```

КОМПОСТЕР                26.03.1992
dim c(126,10),e(126,20),r(9,9),t(9),
    tmp(9),u(9),v(9),x(9),y(9)
input "Input filename"; filename$
open filename$ for input as #1
input #1,n
for i=1 to n: input #1,x(i),y(i):next i
input #1,s
for i=1 to s: input #1,u(i),v(i):next i
close #1
if n < s or s < 2 then
    print "Ошибочные данные": end
for i=1 to n: for j=1 to n
    r(i,j) = (x(i)-x(j))^2 + (y(i)-y(j))^2
next j,i

```

```

'Генерирование конфигураций (сочетаний)
i = 1: t(1) = 0: count = 0
while i > 0
    t(i) = t(i) + 1
    while i < s: i=i+1: t(i)=t(i-1)+1: wend
    while t(i) <= n
        count = count + 1
        for j=1 to s: c(count,j)=t(j): next j
        t(i) = t(i) + 1
    wend
    while t(i) >= n-s+1: i = i-1: wend
wend

```

```

' Поиск одинаковых конфигураций
ii = 0
for i=1 to count
    if c(i,s+1) <> 0 then goto 5
    ii = ii+1: e(ii,1) = 1
    c(i,s+1) = -1: jj = 1
    for j=i+1 to count
        if c(j,s+1) <> 0 then goto 4

```

```

for p=1 to s: t(p) = 0: next p
for k=1 to s
    for p=1 to s
        if t(p) = -1 then goto 2
        for q=1 to s
            tmp(q) = r(c(j,p),c(j,q)): next q
        for l=1 to s
            for q=1 to s
                if r(c(i,k),c(i,l))=tmp(q) then
                    tmp(q) = -1: goto 1
                end if
            next q
        1 next l
        sum = 0
        for q=1 to s: sum = sum+tmp(q): next q
        if sum = -s then t(p) = -1: goto 3
    2 next p
    3 next k
    sum = 0
    for k=1 to s: sum = sum + t(k): next k
    if sum = -s then
        c(j,s+1) = -1
        jj = jj+1
        e(ii,jj) = j
    end if
    4 next j
    e(ii,20) = jj
    5 next i

```

```

' Перебор вариантов нажатий
kcomp = 0: nmaxglob = 0
for i=1 to ii'Конфигурации
    pmin = n: nmax = 0
    for j=1 to 2^e(i,20)-1'Вариант нажатия
        b$ = bin$(j): p = 0
        for k=1 to n: t(k) = 0: next k
        for k=1 to e(i,20)
            if mid$(b$,k,1) = "1" then
                p = p + 1
        for l=1 to s: t(c(e(i,k),l))=1: next l
        end if
        next k
        sum = 0
        for k=1 to n: sum = sum + t(k): next k
        if sum = n and p < pmin then pmin = p
        if sum > nmax then nmax = sum
        next j
        if nmax = n then kcomp = kcomp + 1
        if nmax > nmaxglob then
            nmaxglob = nmax
            compnmax = e(i,1)
        end if
        if nmax = n then
            print "Пункт 4. Компостер:";
            for j=1 to s: print c(e(i,j),j);:next j
            print "Минимальное число нажатий=":pmin
            end if
        next i
        print "Пункт 2. К =": kcomp
        if kcomp = 0 then
            print "Пункт 3. Компостер:";
            for j=1 to s
                print c(compnmax,j);: next j
            print "Максимальное число отверстий =":
                nmaxglob
        end if
    end
end

```

Командный турнир школьников по программированию

Осенью 1991 г. прошел Второй всесоюзный командный турнир АУИформ по программированию для школьников. К концу сентября в оргкомитет прислали заявки на участие в турнире более 70 команд, и 8 октября все они имели возможность участвовать в заочном отборочном туре. Команде из четырех учащихся одной школы, ПТУ или кружка предлагалось решить с помощью компьютера за 8 ч следующую задачу (авторы А. Гейн, А. Сенокосов, М. Алексеев).

Жители города Яшин-Заречный давно беспокоит проблема плохой работы основного автобусного маршрута «Автаостанция» — «Больница». Избиратели потребовали от городских властей ее срочного решения, и власти обратились за помощью к квалифицированным программистам, объявив открытый конкурс на создание лучшей программы планирования автобусного расписания, пригодной, может быть, и для других маршрутов.

Исходные данные. Начало движения — в 6 ч, окончание — в 24 ч. Это время разбито на пять периодов с различной частотой движения автобусов:

с 6 до 7 автобусы должны приходить каждые 10—12 мин;

с 7 до 10 — каждые 7—9;

с 10 до 16 — каждые 10—12;

с 16 до 19 — каждые 7—9;

с 19 до 24 — каждые 10—15 мин.

Продолжительность рейса от А до Б и от Б до А не зависит от времени суток и равна Т.

Рабочий день водителя — 8 ч (с обедом) плюс-минус разумный допуск с учетом продолжительности рейса. Часовой обеденный перерыв может быть не ранее чем через 3 ч и не позднее чем через 5 ч после начала работы. Кроме этого, между рейсами на конечных остановках автобус должен постоять не менее 10 мин. Для простоты принимается, что за каждым автобусом закреплен только один водитель.

Требуется сделать программу, которая, получив на входе Т, составит:

1) расписания отправления автобусов данного маршрута с конечных остановок А и Б (для пассажиров);

2) диспетчерское расписание движения для всех автобусов данного маршрута.

Предполагается, что необходимое количество автобусов и водителей N определяется в ходе решения задачи.

По желанию программистов допускаются (поощряемые дополнительно) усложнения постановки задачи.

Например, ближе к жизни ситуация, когда число автобусов М меньше числа водителей N. При этом кроме графиков движения автобусов придется составлять и графики работы водителей.

Кроме того, продолжительность рейса обычно зависит от времени суток и вместо одной величины Т можно использовать Т₁, Т₂, Т₃, Т₄, Т₅.

Подобные и прочие допущения, принимаемые программистами по своему усмотрению в сторону усложнения или упрощения задачи, должны быть явно оговорены.

50 команд прислали свои решения. По результатам их тщательной проверки на финальный тур были приглашены 12 лучших команд: Бийск, Дом технического творчества, руководитель Гопкало С. В., Стригин В. Л.; Волгоград, школа № 78, Лепехин Ю. В.; Донецк, технический колледж, Федоров Г. И.; Екатеринбург, лицей при УрГУ, Филмоненков Д. О.; Екатеринбург, школа № 45, Степанова Н. В.; Миасс, школа № 18, Алексеева Т. М.; Москва, школа № 57, Зарубина Е. А.; Орша, школа № 2, Борисова Т. А.; Санкт-Петербург, физико-математическая гимназия № 30, Смирнова Е. А.; Чебоксары, школа № 27, Шипилевский А. С.; Челябинск, физико-математический лицей № 31, две команды, Погодин А. П.

Финальный тур прошел 30 ноября — 2 декабря в Миассе, где смогли собраться 10 из 12 приглашенных команд. Командам были предоставлены компьютеры IBM PS/2, дискеты с разработанной В. Прохоровым системой «Пифагор» и следующее задание.

Надеемся, игра «Тетрис» вам хорошо известна. Программный комплекс «Пифагор» мало известен. Вам потребуется его изучить самостоятельно с помощью встроенных подсказок, файлов документации и примеров.

Требуется разработать с помощью комплекса «Пифагор» программную реализацию игры «Тетрис», которая имела бы три режима работы:

а) традиционный, в котором компьютер «выбрасывает» фигуры, а играющий управляет их укладыванием;

б) нетрадиционный, в котором фигуры «выбрасывает» играющий, а укладывает

их компьютер;

в) демонстрационный, в котором компьютер играет сам с собой, укладывая случайные выпадающие фигуры.

Программа должна запрашивать параметры игры (размера «стакана» и что еще сочтете необходимым), показывать на экране ход игры в привлекательной форме, регистрировать количество «съеденных» слоев, обладать чертами, делающими работу приятной и удобной.

В равных условиях оказались и искусственные знатоки IBM и закаленные борцы с «Агатом» и УКНЦ. Полтора дня работали команды над этим заданием, а затем сдавали свои работы, а точнее — защищали их перед жюри, как это водится на конкурсах и конференциях. При этом удачно совместились проверка, апелляция и научные споры. Практически все команды разобрались с новой системой и в той или иной степени справились с первым пунктом задания, а алгоритм укладки фигур компьютером излагали теоретически.

Юные программисты состязались также в личном блиц-туре, где за час на бумаге нужно было решить следующую задачу (авторы Е. Каптив и В. Цикоца).

Имеется цепочка символов ограничен-

ной длины, состоящая из букв латинского алфавита.

Можно выполнять свертку повторяющихся подстрок этой строки по следующим правилам:

а) несколько последовательных повторений одной и той же подстроки заменяются так: aaa на $3(a)$, $abcdcd$ на $a2(bc)d$;

б) правило а можно применять многократно к уже свернутой строке, например $2(a)b2(a)b$ заменяется на $2(2(a)b)$.

Требуется программа, выдающая свертку для введенной строки.

Победительницей Второго всесоюзного командного турнира школьников по программированию стала команда № 1 из лицея № 31 Челябинска, на втором месте — победители первого турнира, школа № 27 из Чебоксар, третье место не присуждалось.

Остается добавить, что организовал и провел турнир Центр информационных технологий Миасского горно, а финансировало турнир главно Челябинского обл. исполкома.

Заявки на участие в следующем турнире можно направлять в Центр ассоциации учителей информатики по адресу: 660074, Красноярск-74, а/я 24160.

110

Мне сверху видно все

Неизвестно, что этому причиной — присущая человеку потребность в общении или действительно необходимость в обмене информацией, но рано или поздно владелец любого, даже самого персонального, компьютера начинает подумывать о включении его в сеть. Это можно проделать разными способами: от прокладки обычного провода до ближайшего компьютера до хулиганского вторжения в глобальные системы связи.

Но любое подключение требует скрупулезного изучения принципов работы, а главное, способов кодирования информации в системе, что не только расширяет кругозор, но и отнимает много времени, причем не только у хулиганов.

ЧТО? МОЖЕТ ЭВМ

Сберечь время и силы позволит новый стандарт компьютерной сети «Е-Систем». Он поддерживает любые каналы — от локальных сетей до глобальных коммуникаций, позволяя связать между собой компьютеры, находящиеся на земле, в океане, в воздухе и даже в космосе. А совместимость «Е-Систем» с космической системой АВАКС позволит ее пользователям, сидящим дома за дисплеями своих компьютеров, с полным правом промурлыкать: «Мне сверху видно все, ты так и знай».

Загнивает Запад...

Одно время по американским ПК «ходила» программа с завлекательным названием «Rockvideo». Запустив ее, владелец компьютера мог наслаждаться мультфильмом, посвященным звезде эстрады и кино Мадонне; однако кончался мультик сообщением «Только идиот использует компьютер для рассматривания кинозвезд!» и... уничтожением всех файлов на дисках.

Мораль этой «электронной басни» действовала недостаточно. Фирма IBM выпускает в продажу набор «IBM PS/2 TV» (плата видеоадаптера, динамик, телетюнер, программное обеспечение), позволяющее смотреть телепрограммы (до 70 одновременно — вот извращенцы!) на экране компьютера PS/2.

Т. ЮРЬЕВА

Бизнес: что есть что?

Ведение бизнеса в той или иной стране, несомненно, определяет его специфику. Различают «американский», «японский», «латиноамериканский» бизнес и т. д. Надеемся, что в ближайшее время достойное место в мировом бизнесе займет и российский бизнес. В то же время, несмотря на существующие различия между теми или иными моделями бизнеса, можно выделить общие закономерности его становления и развития, имеющие общеэкономический аспект.

В настоящее время в большинстве стран мира бизнес изучают не только в высших, но и в средних учебных заведениях. Для этих целей используются специальные учебники, многие из которых ориентированы на компьютеризированный курс обучения. В нашей стране до недавнего времени об изучении такой дисциплины, как основы бизнеса, не могло быть и речи. Сейчас ситуация меняется. Однако, поскольку проблемы бизнеса в нашей стране не рассматривались в течение длительного периода ввиду отсутствия реальных рыночных отношений, соответствующая учебно-методическая литература по данным вопросам практически не издавалась.

Ныне существующий пробел ликвидируется. Появляются работы, полностью посвященные теории бизнеса, методике его преподавания. В то же время до сих пор ощущается недостаток в учебно-методических пособиях, предназначенных для преподавателей и учащихся средних учебных заведений. А ведь именно здесь формируются основы экономического мышления будущих предпринимателей. Актуальным представляется использование знаний по основам бизнеса в процессе преподавания курса «Информатика» в средних учебных заведениях.

В процессе разработки курса автор использует накопленный опыт как в теоретических вопросах, так и в методике преподавания бизнеса в странах рыночной экономики, но в то же время полностью его не дублирует.

Целью курса «Бизнес: что есть что?» является формирование теоретических знаний у преподавателей и учащихся средних учебных заведений в области бизнеса. Курс апро-

бировав автором в процессе преподавания в «Школе бизнеса», созданной на базе одной из московских средних школ.

Курс «Бизнес: что есть что?» состоит из вводной части и шести глав.

В первой главе курса «Бизнес есть бизнес» (Business is business) дается определение бизнеса, рассматриваются цель и средства достижения, предлагается краткий исторический экскурс становления и развития мирового бизнеса, включая российский бизнес.

Во второй главе «Фирма — экономическая клеточка бизнеса» исследуются различные виды фирм в зависимости от формы собственности, а также количественных характеристик (мелкий, средний, крупный бизнес).

В третьей главе «Конкуренция — главный двигатель бизнеса» показывается роль конкуренции в рыночной экономике, рассматриваются различные виды конкуренции: совершенная, несовершенная, ценовая, неценовая и др.

В четвертой главе «Потребитель — король рынка» раскрывается роль потребителей в современном рыночном хозяйстве.

В пятой главе «Бизнес: условия его ведения» анализируется деятельность учреждений, обслуживающих бизнес и приводящих его в движение. Это биржи, банки, органы ресурсного обеспечения и др.

В шестой главе «Бизнес» традиционный и новый» представлены конкретные направления бизнеса. Особое внимание обращается на его новые виды (информационный бизнес, рисковое предпринимательство и др.).

Каждый раздел курса включает краткий словарь ключевых экономических терминов, которые даются как на русском, так и на английском языке. С некоторыми из них читатель может познакомиться уже сейчас.

Аудит (Audit) — детальное исследование финансово-хозяйственной деятельности компаний. Цель аудиторской работы — дать заключение о финансовом положении проверяемой компании. Как правило, она проводится перед ежегодным собранием акционеров. Многие аудиторы оказы-

вают также консультационные услуги по совершенствованию финансово-хозяйственной работы компании.

Бизнес (Business) — экономическая деятельность, имеющая своей целью получение прибыли. Совокупность субъектов (предприятий, фирм), функционирование которых ориентировано на максимизацию прибыли, составляет в условиях рыночного ведения хозяйствования коммерческий сектор экономики.

Брокер (Broker) — юридическое лицо, которое выполняет роль посредника между покупателями и продавцами ценных бумаг, валюты, новых технологий и др.

Брокеридж (Brokerage) — вознаграждение, которое получает брокер за посредническую деятельность между покупателями и продавцами. Оно может устанавливаться в форме фиксированного процента от стоимости сделки или в виде определенной суммы.

Видимая торговля (Visible Trade) — торговля товарами в материально-вещественной форме (продовольствие, машины, сырье и т. д.).

Двойная цена (Dual Price) — это положение, когда один и тот же товар (услуга) имеет двухъярусную цену. Он может быть продан по более низкой цене на контролируемом рынке и по высокой цене на открытом рынке. В целях социальной защиты определенных слоев населения в ряде стран товары первой необходимости (хлеб, сахар и др.) реализуются по фиксированным, достаточно низким ценам посредством специализированных магазинов. Одновременно эти же товары можно приобрести по более высоким ценам на открытом рынке.

Дебитор (Debtor) — юридическое или отдельное лицо, имеющее задолженность конкретному предприятию, организации, государству, физическому лицу.

Демпинг (Dumping) — продажа товаров по бросовым ценам, которые, как правило, ниже издержек производства. Используется для избавления от излишних товарно-материальных запасов, завоевания рынков сбыта, установления новых торговых связей.

Дуополия (Duopoly) — ситуация на рынке, когда только два продавца предлагают идентичную продукцию (в противоположность монополии, когда существует один продавец). В ряде случаев дуополия рассматривается как одна из форм олигополии или ее ограниченный вариант.

Инжиниринг (Engineering) — оказание инженерно-консультационных услуг, связанных с разработкой процесса производства, его обеспечения, реализации полученной продукции.

Инновация (Innovation) — новшество, нововведение. Означает появление новых товаров и услуг на рынке или улучшение качества уже суще-

ствующих; введение новых методов производства; открытие новых рынков; создание новых типов организации и др.

Консалтинг (Consulting) — консультирование фирм в области теории и практики бизнеса.

Кредитор (Creditor) — тот, кто предоставляет деньги или товар в долг, как правило, с уплатой процента. В качестве кредитора могут выступать частное лицо, банк, фирма, компания, государство.

Кредитная карточка (Credit Card) — документ, дающий его владельцу право на кредит. В настоящее время широко распространена в США и других развитых странах.

Лизинг (Leasing) — предоставление в аренду оборудования, транспортных средств и т. д. без перехода права собственности.

Лот (Lot) — стандартная по количеству и качеству партия товара; группа товаров, продаваемых как единое целое.

Невидимая торговля (Invisible Trade) — торговля услугами банков, судоходных компаний, туристических организаций и др.

Олигополия (Oligopoly) — ситуация на рынке, когда число продавцов идентичных товаров и услуг ограничено, но больше, чем один (монополия) или два (дуополия).

Открытая экономика (Open Economy) — экономика, функционирующая без государственного контроля. По отношению к внешней торговле это означает, что экспорт и импорт товаров и услуг осуществляются исходя из интересов производителей и потребителей, обеспечивается свободное движение труда и капитала (в противоположность изолированной или закрытой модели экономики).

Приватизация (Privatization) — процесс денационализации (продажи, передачи) государственной собственности (предприятий, жилых зданий и т. д.) в частную собственность.

Равновесная цена (Equilibrium Price) — цена, при которой спрос и предложение товаров и услуг равны. Изменение спроса или предложения приводит к колебаниям равновесного продукта и цены. Равновесная цена может быть мгновенной, краткосрочной и длительной.

Рынок наличного товара (Spot market) — рынок, на котором ведется купля-продажа реальных товаров (в противоположность фьючерскому рынку).

Рынок фьючерский (Future Market) — рынок, на котором производится покупка и продажа товаров, валюты, ценных бумаг и т. д. исходя из их предполагаемой цены или курса на определенную дату в будущем.

Смешанная экономика (Mixed Economy) — тип экономики, соединяющей в себе различные формы хозяйствования и организации экономики.

Фирма «ЛИнТех» представляет новейшие сетевые системы для КУВТ «КОРВЕТ», «НЕЙВА», «ФОРМАНТА», «ОРБИТА».

В настоящий момент существуют три модификации сетевой системы NET-CP/M:

NET-CP/M A-1.0 — сетевая версия ОС CP/M для КУВТ стандартной конфигурации. Позволяет на РМУ работать в ОС CP/M, обращаясь к дискам РМП. Все возможности стандартной CP/M сохранены. Отличительные особенности — мощная система разграничения доступа, высокая скорость обмена по сети (при стандартном сетевом оборудовании), стопроцентная защита от сбоев сети и дисков, высокие сервисные возможности. Предназначена для ведения учебного процесса.

NET-CP/M B-1.0 — многопользовательская операционная система с головной машиной IBM PC/XT/AT/386/486 и терминалами «КОРВЕТ» (до 64 мест), снабженными высокоскоростными сетевыми адаптерами (375 Кбод). В настоящее время разработаны мощная обучающая система, сетевая СУБД, табличный и текстовый редакторы, электронная почта, подготавливаются учебные курсы по ряду предметов. Система открыта для развития приложений на базе единого графического интерфейса. Возможность работы со стандартными программами CP/M сохранена в файловой системе MS-DOS IBM PC. Предназначена для использования в качестве общеуниверситетской (в том числе и на крупных промышленных предприятиях) локальной сети, совмещающей обработку деловой информации и процесс обучения, представляет собой качественно новую систему, соответствующую мировому уровню развития сетевых технологий.

NET-CP/M AB-1.0 — промежуточный этап между A-1.0 и B-1.0 — представляет собой NET-CP/M A-1.0 с установленными высокоскоростными (375 Кбод) сетевыми адаптерами. Эта система позволяет с минимальными затратами заменить впоследствии головную машину на IBM-совместимую и получить NET-CP/M B-1.0.

Наряду с сетевыми системами фирма «ЛИнТех» поставяет разнообразные программные продукты для вышеназванных КУВТ, отличающиеся высочайшим качеством исполнения, обилием графики, высокой надежностью.

Ассоциация «ЛИнТех» продолжает регистрацию пользователей упомянутых КУВТ, объявленную в ИНФО 1-92. Ассоциация ставит целью объединение пользователей для максимального удовлетворения их потребностей в программном и аппаратном обеспечении, создание условий для обмена информацией. Для членов Ассоциации будет функционировать «горячая линия» — по телефону мы сможем ответить на вопросы по программному и аппаратному обеспечению нашей фирмы.

С января 1993 г. фирма «ЛИнТех» начинает издавать компьютерный журнал, ориентированный на пользователей компьютеров типа «Корвет», пилотный номер которого планируется выпустить в конце 1992 г. Кроме того, члены Ассоциации могут получить бесплатный пакет игровых программ при условии оплаты дискет и почтовых расходов. Для регистрации достаточно вырезать, заполнить и выслать в наш адрес анкету. Адрес: 119501, Москва, а/я 942. Телефон: (095) 273-50-14. Телефакс: (095) 292-65-11 LINTEN Вох 9318.

РЕГИСТРАЦИОННАЯ АНКЕТА УЧАСТНИКА АССОЦИАЦИИ «ЛИнТех»

Адрес _____

Телефон _____ Телефакс _____

Название организации _____

ФИО ответственного лица _____

Используемая вычислительная техника _____

Используемое программное обеспечение _____

Анкета должна быть заверена круглой печатью вашей организации.

Научно-производственная фирма «Фарматон»

предлагает

Сетевая студия (сетевой пакет для учебного класса)

Пакет предназначен для работы в классе на основе ПЭВМ «Искра-1031» или 1030, объединенных в сеть по линиям ИРПС (адаптер АПД).

Программа, функционирующая на центральной машине (рабочее место преподавателя — РМП), обеспечивает управление периферийными машинами (рабочими местами учащихся — РМУ), позволяя преподавателю:

- запускать на РМУ программы, расположенные на накопителях РМП или РМУ;
- организовывать очереди последовательно выполняющихся программ для каждого РМУ;

- прерывать выполнение программ;
- переносить файлы между РМП и РМУ;
- просматривать экран и директории РМУ.

При использовании учебных программ, специально написанных для пакета, становится возможным также:

- в любой момент узнать состояние программы на РМУ (процент выполнения задания, текущая оценка, любые другие сведения);

- автоматически по завершении программы принять оценку, записать ее в базу данных и распечатать;

- просматривать список программ (с оценками), выполненных каждым учащимся.

Пакет *сетевая студия* — NET STUDIO автоматизирует всю операторскую работу в сети. Достаточно в начале сеанса работы с классом сформировать очереди программ, и весь сеанс будет проведен автоматически. Преподавателю не придется ходить по классу с дискетами и журналом.

Основная управляющая программа (NS) оснащена удобным и простым многооконным интерфейсом. Вы управляете классом с помощью системы меню или функциональных клавиш. Те, кто работал с Norton Commander, освоят NS мгновенно.

Программа постоянно отображает текущее состояние всех РМУ. Можно управлять каждой машиной в отдельности или взаимодействовать сразу с группой машин. Меню заданий упрощает выполнение повторяющихся команд и последовательностей — вы можете сформировать его сами.

NS надежно держит связь — любая из машин может быть выключена и включена снова в любой момент, даже центральная!

В состав пакета входят программы для проведения тестового контроля — с их помощью быстро подготавливаются сложные контролирующие-обучающие модули, полноценно работающие в сети и автономно. Вопросы и варианты ответов для них набираются с помощью любого текстового редактора. В программе могут использоваться графические диаграммы, создаваемые графическим редактором.

Пакет снабжен документацией и фрагментами программ, при помощи которых можно дополнить программы, написанные на Си, Турбо-Паскале или Бейсике, сетевыми средствами.

В состав пакета входят драйверы клавиатуры, принтера, экрана для «Искры-1030/31».

Характеристики:

число обслуживаемых периферийных машин — до 12;

скорость обмена в сети — до 19200 бод;

формат базы данных — совместим с FoxPro/FoxBase.

Пакет будет адаптирован для классов и сетей других типов — на основе ПЭВМ ЕС 1841 (сеть «Сатурн»), IBM PC-совместимых.

Один пакет для одного класса. Поставка на защищенных от копирования дискетах 360 К.

Наши реквизиты: почтовые — 656099, г. Барнаул, а/я 3081, НПФ «Фарматон»; тел. (385-2)-26-07-58;

банковские — р/с 467875 в Агропромбанке г. Барнаула, МФО 101017.

В. ОНЕГОВ

Проблемные задачи на уроках информатики

Школьный курс ОИВТ представляет собой ярко выраженный общеобразовательный предмет. Его положение среди дисциплин учебного плана можно сравнить лишь с положением философии в общей системе наук [1]. В силу этого с неизбежностью курс обладает (должен обладать) широкими межпредметными связями, в которых следует различать два аспекта.

Первый — знания, умения и навыки, приобретенные в процессе изучения ОИВТ, в дальнейшем будут необходимы при изучении других школьных предметов (по мере компьютеризации общества и образования).

Второй аспект выражается в том, что с введением в школу ОИВТ стало возможным формирование у учащихся представлений об этапах полного решения любой практической задачи. Второй аспект связан с методологией науки «информатика». Нам представляется совершенно необходимым рассмотрение на уроках ОИВТ в школе задач, позволяющих последовательно применять эту методологию. В этом отношении межпредметные связи во втором аспекте представляются «центром» ОИВТ. Важно не только научить учащихся составлять достаточно сложные, но искусственные алгоритмы, не только привить операционные навыки работы с простейшим программным обеспечением (текстовый редактор и др.), не только познакомить с алгоритмическими языками высокого уровня, но и дать представление о месте информатики в системе всех наук, всех видов общественно значимой деятельности. Сделать это можно только на примерах задач из различных областей знаний, причем, чем шире спектр примеров, тем более глубоким будет представление о месте информатики. Во взглядах на школьную информатику автор во многом солидарен с идеями, высказанными в [2].

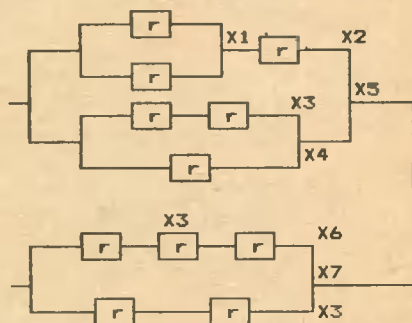
Не является большим откровением, что подбор содержательных примеров, на которых можно было бы показать все необходимые этапы решения любой прикладной задачи, причем в приемлемое учебное время, есть дело само по себе не легкое. Многие учителя испытывают в этом отношении известный методический голод и неуверенность.

В статье предлагается несколько задач такого типа, апробированных на уроках ОИВТ в средней школе. Хочется отметить, что все уроки, на которых решались эти задачи, проходили в атмосфере творческого подъема, повышенной организованности и искреннего интереса.

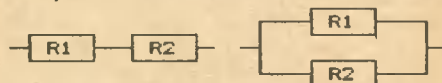
Задача 1.

Нахождение функциональной зависимости сопротивления электрической цепи от сопротивления участка цепи.

Постановка. Исследовать зависимость сопротивления R электрической цепи (рис. 1) от сопротивления r участка цепи. Результатом исследования должно быть нахождение функциональной зависимости $R=R(r)$.



Математическая модель. В основу расчета рассматриваемой электрической цепи положим закон Ома для участка цепи, состоящей из двух сопротивлений, соединенных последовательно (рис. 2) и параллельно (рис. 3).



Сопротивление электрической цепи с последовательно соединенными сопротивлениями R_1 и R_2 определяется формулой $R=R_1+R_2$. А в случае параллельного соединения — $R=R_1 \times R_2 / (R_1 + R_2)$.

Расчет сопротивления R всей электрической цепи можно свести к неоднократному расчету участков цепи, состоящих из двух сопротивлений, соединенных последо-

тельно или параллельно. На рис. 1 через X_k обозначены сопротивления, на которые заменяются сопротивления соответствующих участков цепи.

Алгоритм. В связи с выбранным методом расчета сопротивления всей цепи предусмотрим два вспомогательных алгоритма-функции с именами ПОС (R_1, R_2) и ПАР (R_1, R_2).

```

алг ПОС ( арг вещ R1, R2)
нач
    знач:=R1+R2
кон
алг ПАР ( арг вещ R1, R2)
нач
    знач:=R1*R2/(R1+R2)
кон
    
```

Будем строить таблицу значений сопротивления всей цепи R при N значениях r сопротивления участка цепи ($r=h_i, i=0, 1, \dots, N$). Общий алгоритм построения такой таблицы таков:

```

алг ЦЕПЬ ( арг вещ h, арг цел N,
    рез вещ таб R[0:N])
надо | вычислить таблицу
    функции R=R(r) и
    напечатать коэффициенты
    k=R/r
нач цел i, вещ r, X1, X2, X3, X4, X5,
    X6, X7, k
r:=0; R[0]:=0
нц для i от 1 до N
    r:=r+h; X1:=ПАР(r, r);
    X2:=ПОС(r, X1);
    X3:=ПОС(r, r);
    X4:=ПАР(X3, r);
    X5:=ПАР(X2, X4);
    X6:=ПОС(X3, r);
    X7:=ПАР(X6, X3);
    R[i]:=ПОС(X5, X7) k:=R[i]/r
вывод нс, "k=", k
кц
кон
    
```

Реализация на ЭВМ. Алгоритм, записанный на школьном алгоритмическом языке, при работе на ПЭВМ вводится в виде Бейсик-программы, причем на экране дисплея ограничивается вывод графика функции $R=R(r)$ по значениям построенной таблицы.

Результаты и выводы. График функции $R=R(r)$ представляет собой прямую. Соответственно все найденные значения k совпадают. Вывод: искомая функциональная зависимость — прямая пропорциональность с коэффициентом пропорциональности k , т. е. $R=kr$.

Замечание. Априорный ответ на вопрос о функциональной зависимости сопротивления электрической цепи от сопротивления участка цепи вызывает затруднения и у более искушенных в физике людей, чем старшеклассники. Возможное доказательство прямо

пропорциональной зависимости методом индукции может представлять интерес как иллюстрация применения метода полной математической индукции, кстати не изучаемой в школе.

В этом отношении рассмотренная задача действительно является примером компьютерного решения содержательной задачи.

Задача 2.

Разработка имитационной модели-игры «Пушка — цель».

Постановка. В некоторой точке местности занимает позицию пушка, ствол которой ориентирован в направлении цели, имеющей прямоугольную конфигурацию. Цель подвижна и может перемещаться вперед и назад вдоль прямой, соединяющей цель и пушку. Наименьшее расстояние цели от пушки — S_{min} , наибольшее — S_{max} . Таким образом, расстояние цели от пушки S удовлетворяет неравенству $S_{min} < S < S_{max}$. Пусть высота цели равна h , а длина — m . Будем считать, что задана V_0 — начальная скорость вылета снаряда из ствола пушки. Обозначим α — угол прицела (угол наклона ствола к горизонту), ясно, что $0 < \alpha < 90$.

В этих условиях разработать имитационную модель полета снаряда, выстреленного из пушки в направлении цели под углом α к горизонту. При этом на экране дисплея в соответствующем масштабе должны быть отражены: положение пушки, наклон ствола, положение цели, траектория полета снаряда.

Работа с моделью должна проходить в виде игры: при каждом появлении цели на расстоянии S от пушки (значение S выбирается случайным образом) учащемуся разрешается сделать не более трех «выстрелов» при различных значениях угла прицеливания α , причем модель должна предусматривать случайные воздействия на дальность полета снаряда ($\pm d$), при условии, что $d < 0.01S$. Выигрыш достигается, если хотя бы один из «выстрелов» в трех попытках приводит к попаданию в цель.

Математическая модель. Математическую модель движения снаряда, выпущенного из ствола пушки с начальной скоростью V_0 под углом α к горизонту, будем строить в предположениях:

1. На снаряд в процессе полета действует сила притяжения Земли.
2. Поверхность Земли считаем плоскостью.
3. Введены взаимно перпендикулярные оси координат OX и OY с началом в центре пушки. Ось OX ориентирована вдоль прямой пушка — цель. Ось OY перпендикулярна поверхности Земли.
4. Суммарный результат всех неучтенных воздействий на снаряд в полете будем учи-

тывать в виде случайной величины d , влияющей на дальность полета ($\pm d$), причем $d < 0.01S$.

Учитывая только предположения 1—3, приходим к двум формулам, определяющим координаты траектории полета снаряда в выбранной системе координат.

$$X = V_0 \cdot \cos(\alpha) \cdot t, \quad Y = V_0 \cdot \sin(\alpha) \cdot t - gt^2/2 \quad (1),$$

где t — время в секундах, g — ускорение свободного падения.

Из этих формул непосредственно находится время полета снаряда до падения t_0 и L — теоретическую дальность полета:

$$t_0 = 2 \times V_0 \cdot \sin(\alpha) / g \quad \text{и} \quad L = V_0^2 \times \sin(2\alpha) / g.$$

Кривая, описываемая уравнениями (1), является параболой с вершиной при $X = L/2$. Из практики известно, что действительная траектория полета снаряда несколько отличается от параболы, причем левая ветвь сходна с ней, а отличие правой — достаточно заметно. Причиной этого являются не учтенные в предположениях 1—3 влияния на полет снаряда (например, ветер). Поэтому в математической модели все неучтенные влияния совокупно учитываются в условии 4.

При учете условия 4 в математической модели полета снаряда примем следующую гипотезу: 1) при X меньших $L/2$ (левая ветвь) траектория описывается уравнениями (1); 2) при X больших $L/2$ (правая ветвь) траектория описывается следующими уравнениями:

$$X = V_0 \times \cos(\alpha) \cdot t + (t - t_0/2) \times b, \\ Y = V_0 \cdot \sin(\alpha) \times t - gt^2/2,$$

где $b = \pm 2d/t_0$.

Компьютерная модель. Разработанная математическая модель полета снаряда, выстреленного из пушки под углом α к горизонту, служит основой для построения компьютерной модели. На экране дисплея в выбранном масштабе нужно отобразить положение пушки, ее ствола, положение цели на расстоянии S от пушки и, наконец, траекторию полета снаряда. Параметры V_0 , S_{max} , S_{min} , h , m задаются в самом начале программы. При этом если выбрать $V_0 = 48$ м/с, $S_{max} = 245$ м, $S_{min} = 125$ м, то получается удобный масштаб: 1 пиксел соответствует 1 м. Затем случайным образом выбирается расстояние от пушки до цели: $S_{min} < S < S_{max}$. На этом расстоянии на экране появляется цель (прямоугольник со сторонами h и m).

Учащемуся предлагается ввести угол α для возможного поражения цели. При этом ствол пушки принимает положение, соответствующее углу α . Вырабатывается случайное воздействие на дальность полета $\pm d$.

На экране дисплея изображается траекто-

рия полета снаряда в соответствии с уравнениями, полученными в математической модели. При этом проверяется условие пересечения траектории снаряда границы цели, т. е. одновременное выполнение условий $S < X < S + m$ и $Y < h$. Если пересечение есть, то построение траектории прекращается — цель поражена. В противном случае продолжается построение траектории полета снаряда до пересечения с поверхностью Земли, после чего модель возвращается в состояние, в котором она находилась перед вводом угла α .

Возможны три попытки. Если ни одна не принесла успеха, то модель задает следующее положение цели, т. е. новую серию «выстрелов».

Бейсик-алгоритм. Составлен для $V_0 = 49$ м/с.

```

10 REM Компьютерная модель:
20 REM ПУШКА - ЦЕЛЬ
30 CLS
40 INPUT "Введите длину цели m":M
50 INPUT "Введите высоту цели h":H
60 SCREEN 2: COLOR 15,4
70 OPEN "GRP:" AS#1
80 SCREEN 2
90 LINE (13,178) - (255,178),15
100 LINE (10,185) - (15,175),7
110 LINE (10,175) - (15,165),7
120 LINE (10,185) - (10,175),7
130 LINE (15,175) - (15,165),7
140 PAINT (13,170),7
150 CIRCLE (5,181),4,7
160 PAINT (5,181),7
170 CIRCLE (10,171),4,7
180 PAINT (10,171),7
190 LINE (5,181) - (10,171),7
200 PAINT (9,178),7
210 J=7: A1=45*3.14/180
220 GO SUB 660
230 K=0
240 S=INT(115*RND(-TIME)+125)
250 LINE(S,178-H) - (S+M,178),B,BF
260 DRAW"BM10,10":PRINT#1,
      "Введи угол прицела"
270 A$=""
280 FORN=1 TO 2
290 B$=INKEY$
300 IF B$="" THEN 290
310 A$=A$+B$
320 NEXT N
330 A=VAL(A$): A=3.14*A/180
340 IF A>3.14/2 THEN 260 ELSE LINE
(10,10) - (150,18),4,BF:
      DRAW"BM10,10":PRINT#1,"α="A$
350 J=4: GO SUB 660
360 J=7: A1=A: GO SUB 660
370 V0=48
380 T0=2*V0*SIN(A)/10
390 D=INT(.01*S*RND(-TIME)+1)
400 Q=INT(2*RND(-TIME))
410 IF Q=1 THEN D=-D
420 L=V0*V0*SIN(2*A)/10
430 B=2*D/T0
440 Z=178-Y1+3
450 FOR X=X1+8 TO L+D+8 STEP 5
460 IF X<L/2 THEN T=(X-5)/V0/COS(A)
      ELSE T=(B*X/T0+2*(X-5))/
      (V0*COS(A)+B)
470 Y=V0*SIN(A)*T-10*T*T/2

```

```

480 IF X>=S AND X<=S+M AND Y<=H
      THEN 580
490 LINE (X,178-Y) - (X-5,178-Z),1
500 Z=Y
510 NEXT X
520 LINE(10,10) - (45,18),4,BF
530 DRAW "BM10,10": PRINT#1,
      "НЕ ПОПАЛ!"
540 K=K+1
550 IF K>=3 THEN 610
560 TIME=0
570 IF TIME > 300 THEN LINE(10,10)-
      (164,18),4,BF:
      GO TO 260 ELSE 570
580 SCREEN 0
590 LOCATE10,10: PRINT"ПОПАДАНИЕ!"
600 FOR L=0 TO 700 : NEXT L
610 SCREEN 0
620 LOCATE 5,15: PRINT"Хотите еще раз?"
630 LOCATE 5,17: PRINT"нажмите<пробел>"
640 TIME=0
650 IF INKEY$=" " THEN 80 ELSE IF
      TIME < 400 THEN 650 ELSE CLS: END
660 X1=10+20*COS(A1)
670 Y1=178-20* SIN(A1)
680 YS=(Y1-178)/(X1-10)*5+178
690 LINE(15,YS) - (X1,Y1),J
700 LINE(15,YS-1) - (X1,Y1-1),J
710 LINE(15,YS-2) - (X1,Y1-2),J
720 LINE(15,YS-3) - (X1,Y1-3),J
730 RETURN

```

Обсуждение результатов. После специальной серии «выстрелов» учащиеся делают вывод о том, что: 1) наибольшая дальность достигается при угле прицеливания $\alpha=45^\circ$; 2) одинаковая дальность стрельбы достигается при углах $\alpha=45 \pm \beta$, где $\beta < 45$. При этом углу $\alpha=45-\beta$ соответствует настильная траектория, а $\alpha=45+\beta$ — навесная. Можно разнообразить игру введением условия попадания или в переднюю стенку цели, или в верхнюю часть. Тогда очевидным образом изменится условие попадания в цель.

Задача 3.

Шифрование русского текста на «тарабарский язык».

Постановка. «Тарабарским языком» пользовались древнерусские разбойники для сокрытия смысла своих слов от непосвященных.

Этот язык представляет собой своеобразный шифр, заключающийся в замене одних согласных букв русского алфавита другими. Требуется составить программу, способную переводить русские слова в «тарабарские» и наоборот.

Описание метода. Выпишем все согласные буквы русского языка в алфавитном порядке (используем только строчные буквы)

б в г д ж з к л м н п р с т ф х ц ч ш щ

Всего таких букв — 20. Метод шифровки состоит в том, что каждая из согласных букв, находящихся в левой половине списка, заменяется на букву, стоящую в правой, и наоборот. При этом пробелы, знаки препинания и гласные буквы остаются неизменными.

Алгоритм. Вводим литерную переменную $A = \langle \text{бвгджзклмнопрстфхцчшщ} \rangle$. Очевидно, что $\text{длин}(A) = 20$. Пусть литерная переменная X имеет значение некоторой фразы на русском языке. Алгоритм состоит в просмотре слева направо всех символов переменной X . При этом возникают две возможности: очередной символ — согласная буква и очередной символ — гласная буква, или пробел, или знак препинания. В первом случае находится эта согласная буква среди символов переменной A , вместе с ее порядковым номером. Например, ее порядковый номер i . Тогда символ переменной X , соответствующий $A[i]$, заменяется на символ $A[20-i+1]$. Если же очередной символ переменной X не является согласной буквой, то анализируется следующий по порядку символ.

Бейсик-алгоритм.

```

10 REM <ШИФР ПО ТАРАБАРСКИ>
20 A$="бвгджзклмнопрстфхцчшщ"
30 INPUT X$
40 FOR K=1 TO LEN(X$)
50 FOR I=1 TO 20
60 IF MID$(X$,K,1)=MID$(A$,I,1)
      THEN MID$(X$,K,1)=
      MID$(A$,20-I+1,1): GO TO 80
70 NEXT I
80 NEXT K
90 PRINT X$
100 END

```

Работа с алгоритмом. Вводятся различные русские слова и фразы. Устанавливается, что с помощью этой программы можно дешифровать слова и фразы, написанные по «тарабарски», на русский язык.

Задача 4.

Построение имитационной модели «Броуновское движение».

Постановка. В 1827 г. английский ботаник Броун обнаружил, что взвешенные в жидкости частицы совершают хаотическое движение, которое впоследствии назвали броуновским. Построить динамическую имитационную модель этого явления.

Теоретическая модель. Объяснение броуновского движения дается только в рамках молекулярно-кинетической теории. В соответствии с этой теорией броуновское движение — это результат теплового движения молекул жидкости.

Установлено, что:

1. Все молекулы жидкости, в которой взвешены частицы, находятся в постоянном хаотическом движении. Причем, чем выше температура жидкости, тем больше скорость их движения. Было установлено, что $V_{cp} = kT$ (V_{cp} — средняя скорость движения молекул, T — абсолютная температура жидкости, k — коэффициент пропорциональности).

2. Так как броуновская частица во много

раз больше любой молекулы жидкости, то в любой момент времени ее одновременно «бомбардируют» очень много молекул.

3. Чем выше температура жидкости, тем большее число молекул в единицу времени «атакуют» частицу.

4. При хаотическом движении молекул оказывается, что импульсы движения, передаваемые с разных направлений, не одинаковы и поэтому результирующая сила воздействия на частицу не равна нулю. Она и вызывает движение частицы.

5. Хаотичность движения молекул и, как следствие, хаотичность движения броуновской частицы означают, что направление движения частицы в каждый момент наблюдения есть случайная величина (угол α). Длина пробега частицы между очередными сменами направления тоже величина случайная (r).

6. Если вести наблюдение за положением (X_k, Y_k) частицы через равные отрезки времени (dt), то в каждый момент наблюдения координаты частицы определяются по формулам

$$X_k = X_{(k-1)} + r_k \times \cos(\alpha_k); \quad Y_k = Y_{(k-1)} + r_k \times \sin(\alpha_k), \quad (2)$$

при $k=1, 2, \dots, n$, где (X_0, Y_0) — начальное положение.

7. Средняя величина пробега частицы между двумя последовательными наблюдениями тем больше, чем больше T , $\beta = M_m/M_c$ и чем меньше C (где T — абсолютная температура, M_m — масса молекулы жидкости, M_c — масса частицы, C — вязкость жидкости).

Имитационная модель. Состоит в реализации формул (2) и выводе на экран дисплея ломаной, которая получается в результате соединения последовательных положений частицы. Управляющими параметрами должны быть величины T, β, C . Чтобы отразить на экране дисплея скорость перемеще-

ния частицы, будем считать, что время между последовательными наблюдениями обратно пропорционально температуре, т. е. $dt = 1/T$. Экранный эффект — чем больше температура T , тем быстрее должна двигаться частица по экрану дисплея. Кроме того, будем считать, что максимальная длина пробега частицы между двумя последовательными наблюдениями R прямо пропорциональна β и обратно пропорциональна вязкости C , т. е. $R = \beta/C$.

Значения α и r вырабатываются датчиком случайных чисел.

Алгоритм.

```

10 REM <Броуновское движение>
20 CLS
30 INPUT "Введите температуру T";T
40 PRINT "Удбно брать C/P больше 10"
50 INPUT "Введите вязкость C";C
60 INPUT "Введите величину P";P
70 R=C/P
80 IF T>1 THEN DT=1000/T ELSE DT=10000
90 SCREEN 5
100 W=RND(-TIME)
110 K=50
120 X=128:Y=96
130 FOR I=1 TO K
140 A=RND(W)*360
150 A=A*3.14/180
160 P=INT(RND(W)*(R-2))+3
170 XE=X+P*COS(A)
180 YE=Y+P*SIN(A)
190 LINE (X,Y) - (XE,YE),1
200 Y=YE: X=XE
210 TIME=0
220 IF TIME<DT THEN 220
230 NEXT I
240 GOTO240

```

Работа с моделью. Вводя различные значения параметров T, β, c , наблюдать поведение частицы, найти границы значений этих параметров, при которых картина получается наиболее характерной. Сделать выводы, соответствующие основным положениям молекулярно-кинетической теории. Результаты сравнить с известными физическими опытами.

Н. ДЕЛЕКТОРСКИЙ

Коллективные имитационные игры

Проблема, подтолкнувшая авторов к исследованию увлекательной малоосвоенной области моделирования, знакома каждому преподавателю: школьникам неинтересно учиться. Возникает необходимость разработки методов обучения, основанных на конкретной деятельности учащихся, т. е. перехода от пассивного овладения знаниями к активному.

Возможности игр, особенно компьютерных, связанные с этим направлением, известны давно. Как правило, имеются в виду индивидуальные динамические или логические игры, традиционные и нетрадиционные парные игры. При этом практически не подверглись компьютеризации коллективные игры, хотя комплекты вычислительной техники,

устанавливаемые в школах, предоставляют такие возможности.

Нас заинтересовало именно коллективное игровое социальное моделирование, т. е. игры, в которых группа участников имитирует человеческие взаимоотношения в пределах реального социума. Например, достаточно обобщенные модели завода, магазина, собрания общественной организации и т. п. Каждый участник берет на себя определенную социальную роль, и от успешности его действий зависит успех всей группы. Очевидно, такие игры могут решить некоторые педагогические задачи, не доступные играм другого рода: воспитание чувства ответственности, коллективизма, коммуникативных и интерактивных качеств личности, дисциплины совместного труда. Одновременно решаются традиционные игровые задачи: развитие оперативного и логического мышления, способность к абстракции, тренировка памяти и обучение беглому владению вычислительной техникой.

Разработка описанных ниже игровых социальных моделей «Аэропорт» и «Гражданская оборона» началась с желания авторов наполнить «живым» содержанием раздел курса информатики «ЭВМ в современном обществе», используя возможность моделирования на компьютере. Делалась ориентация на педагогические задачи и учитывалась возможность применения игр в других курсах: «Гражданской обороны» — в разделе «Ядерное оружие иностранных армий» предмета НВП, «Аэропорта» — в курсе изучения основ производства.

Неоднократное применение игр в школе показало, что они с успехом выполняют свои задачи, поскольку вызывают устойчивый интерес учащихся и проходят в естественной деловой атмосфере. В перспективе авторам видится методически продуманная система коллективных игровых социальных моделей, охватывающих различные дисциплины и ориентированных на разнообразные педагогические проблемы.

Приведенные краткие описания двух игр для КУВТ-86 (ДВК-БК) скорее показывают потенциал данной области, чем конкретные детали той или иной реализации.

Игровая социальная модель «Аэропорт»

В игре моделируется работа и взаимодействие служб гражданского аэропорта в реальном масштабе времени (40—45 мин).

Центральная роль принадлежит руководителю полетов, остальные службы условно делятся на подчиненные — выполняющие задания руководителя — и полунезависимые — работающие автономно, но сообщающие руководителю о результатах работы. Смысл

игры заключается в правильном и своевременном взаимодействии служб для безаварийного отправления и приема самолетов. С этой целью каждый играющий получает распписание полетов, на мониторе одного из компьютеров отображается время игры. Критерием оценки успешности игры служит своевременность и безаварийность обслуживания самолетов.

Особенностью игры является широкий разброс сложности исполнительских задач, позволяющий дифференцировать работу участников в зависимости от их подготовленности. Интересна также возможность использовать игру в упрощенном варианте — для выработки общих навыков коллективного труда, и полном — для отработки сложных многосвязных взаимодействий в процессе трудовой деятельности. Несколько затрудняет использование игры необходимость кропотливой игровой подготовки (печать расписаний, передача данных по локальной сети и т. п.).

Пакет состоит из 12 Бейсик-программ, файла данных и текстового файла общим объемом 100 килобайт.

Игровая социальная модель «Гражданская оборона»

Игра моделирует действия частей и соединений Вооруженных Сил и гражданской обороны при массированном ядерном нападении. С этой целью выделяется командно-информационная, разведывательно-расчетная и исполнительская группы. Управление всеми службами осуществляется строго централизованно командиром соединения. Необходимо допустить как можно меньший ущерб охраняемому объекту. Выводы об успешности игры делаются на основании карты местности с нанесенными ударами, предоставляемой отделением информации.

К особенностям игры следует отнести высокую интенсивность работы всех играющих, а также ее методическую близость к разделу «Ядерное оружие иностранных армий» курса начальной военной подготовки. Игра не задействует специфические ресурсы КУВТ, поэтому может быть легко адаптирована к другой технике.

В игру входят 10 Бейсик-программ и три текстовых файла общим объемом 60 килобайт.

Этой публикацией мы хотели бы начать сотрудничество со всеми, кто интересуется данной проблемой.

Наш адрес: 123363, Москва, ул. Сходненская, 35, СШ № 114, Творческий Компьютерный Клуб.

«Игротека» в сельской школе

Вопрос изыскания средств для оборудования только что установленного компьютерного класса и на приобретение новых программ волнует очень многих учителей сельских школ. Задача решается просто, если есть богатые шефы, но при нашем положении экономики не каждый возьмется вам помочь. А вложения нужны немалые, если вы решили всерьез посвятить себя информатике.

Мы нашли способ зарабатывать деньги, не вступая в конфликт с инструкциями и законами, что немаловажно, организовав на базе кабинета информатики «игротеку». Кабинет остается в полном техническом порядке, и, кроме учеников, занимающихся информатикой, там никого нет. Доход мы получаем ежемесячно до 500 руб.

Весь секрет в том, что «игротека» состоит из выносных столов с черно-белыми телевизорами, переделанными под мониторы, и самодельных джойстиков. Выносные мониторы, стоящие в фойе, соединены стандартными разъемами с компьютерами БК-0010Ш.

Технология связи проста: подключаешь разъем к компьютеру, загружаешь игровую программу — и играй, но в фойе! В кабинете можно даже отключить монитор этого компьютера. На выносной игровой стол подается звук, изображение и возможность управления. Сложность была со сбором монет, но и ее мы решили.

В телевизор вмонтирован самодельный монетоприемник, связанный с реле времени. Бросил монетку — и играй 1 минуту. А если игра продолжительная и за 1 минуту не окончишь? Мы сделали реле таким образом, что через 1 минуту игра не останавливается, а лишь отключается ее изображение. Быстрей бросай следующую монету — и еще 1 минута твоя.

Теперь финансовая сторона. Касса монетоприемника опечатана, и вскрывает ее комиссия из числа работников школы и бухгалтерии поселкового Совета, куда после подсчета суммы и проведения необходимых отметок в специальном журнале сдаются деньги.

Поскольку мы стоим на балансе поселкового Совета, то он переводит эти деньги на наш спецсчет. С администрацией школы у нас заключен договор, согласно которому накопленная сумма используется только на нужды кабинета информатики.

Нами разработаны образцы требуемых бумаг, чертежи и схемы подключения, а также более подробное описание создания данной «игротеки». Все, кто нуждается в технической помощи, обращайтесь по адресу: 242730, пгт. Дубровка Брянской области, школа № 2, кабинет информатики.

А. ЧАЗОВ

К вопросу об экспертизе ППИС

В «ИНФО» № 6 за 1991 г. помещена статья О. Г. Парсаданова «О пользовательской экспертизе программ». Проблема качественной экспертизы программно-методических продуктов, поступающих в учебные заведения из самых разных источников, является исключительно актуальной. Слишком много плохих поделок появляется сейчас на рынке ППИС, а рядовой учитель да и городской или районный методист крайне ограничены в возможностях их оперативной оценки. Однако вряд ли эта проблема может быть успешно решена на том пути, который предлагается в вышеупомянутой статье.

Действительно, такие выдвигаемые автором характеристики, как использование экранных окон, экранный справочник-подсказка, возможности распечатки, язык

реализации и т. д., далеко не во всех случаях оказываются сколько-нибудь существенными для оценки дидактической ценности экспертируемого средства. Все зависит от того, как именно предполагается использовать это средство в учебном процессе, какую именно нагрузку должно оно нести.

Что же касается предлагаемых О. Г. Парсадановым способов оценки текстовых и графических редакторов, СУБД и электронных таблиц, то против самих этих подходов вряд ли стоило бы возражать... если бы речь шла о самих этих средствах, а не об их использовании в учебном процессе. Так, например, текстовый редактор с весьма ограниченными возможностями, будучи мало пригодным для практической работы, может в то же время оказаться очень подходящим для иллюстрации учащимся общих

принципов работы подобных средств.

Объективную оценку того или иного программно-методического средства невозможно дать в отрыве от предполагаемых способов его использования в учебном процессе. А это в свою очередь зависит и от категорий обучаемых, и от следования той или иной программе и/или методики обучения, и от уровня обеспеченности занятий вычислительной техникой, да и от ряда других причин.

Естественно, что методика оценки самым существенным образом зависит от категории оцениваемого средства. Не имея возможности в короткой заметке охватить все основные подходы к экспертированию, ограничимся лишь примером официального документа (с небольшими купюрами), который, на наш взгляд, был бы уместен при экспертизе ППИС.

ДОГОВОР № ...

Между Комитетом по народному образованию _____ и руководителем экспертной группы _____

Комитет по народному образованию _____ в лице председателя _____, действующего на основании Положения о комитете, именуемый в дальнейшем «Заказчик», с одной стороны, и руководитель экспертной группы _____, именуемый в дальнейшем «Исполнитель», с другой стороны, заключили настоящий договор о нижеследующем:

1. Заказчик поручает, а исполнитель принимает на себя проведение психолого-педагогической и программно-технической экспертизы ППИС под наименованием _____, разработчик _____

2. Заказчик передает исполнителю на время проведения экспертизы:

эксплуатационную документацию на комплекс программных средств;

существующие на момент заключения договора методические материалы для пользователей; программно-информационные средства на машинных носителях в виде, обеспечивающем их использование во всех режимах работы, предусматриваемых в системе;

перечень учебных заведений (название региона), в которых практически используется экспертируемое ППИС;

иные имеющиеся в распоряжении комитета материалы, могущие способствовать объективной оценке указанного ППИС.

3. Исполнитель обязуется использовать перечисленные в п. 2 материалы исключительно для проведения экспертизы, не допуская их несанкционированного распространения, гарантирует их сохранность и возврат.

4. В процессе проведения экспертизы должно быть сделано следующее:

определен перечень моделей ПЭВМ (из числа IBM-подобных, имеющихся в учебных заведениях (название региона), версий операционных систем и сетевых программных средств, обеспе-

чивающих функционирование экспертируемого ППИС;

дана общая оценка качества функционирования составляющих систему программ;

проведен анализ имеющегося опыта практического применения экспертируемого средства; проанализированы функциональные характеристики системы в режимах подготовки и сопровождения обучающихся курсов;

проанализированы психолого-педагогические характеристики системы;

дана оценка соответствия системы действующим санитарно-гигиеническим требованиям;

разработаны итоговые оценки и рекомендации по применению экспертируемого ППИС в подведомственных заказчику учебных заведениях.

5. По окончании работ исполнитель представляет развернутый отчет по всем разделам, перечисленным в п. 4.

6. Исполнитель обязуется закончить работы в срок не более двух месяцев со дня фактического авансирования работ (см. п. 8 настоящего договора) и предоставления всех материалов, перечисленных в п. 2, а заказчик — организовать их приемку не позднее 15 дней с момента представления отчета.

7. Неотъемлемой частью настоящего договора является СМЕТА РАСХОДОВ, составленная применительно к сложившемуся на (дата) уровне цен.

8. Заказчик авансирует проведение работ в размере 50 % от их общей стоимости не позднее 15 дней со дня подписания договора и полностью оплачивает их не позднее 15 дней по завершению приемки.

9. Расчеты с членами экспертной группы и соисполнителями регулируются их соглашениями с исполнителем (руководителем группы) и не являются предметом настоящего договора.

10. За несоблюдение договора стороны несут ответственность в соответствии с действующим законодательством.

Заказчик	Исполнитель
Председатель Комитета по народному образованию	Руководитель экспертной группы

» _____ 199 г. « » _____ 199 г.

Разумеется, далеко не во всех случаях экспертиза должна быть столь фундаментальной. Но в любом случае она должна давать оценку применимости исследуемого средства в учебном процессе, исходя именно из целей и задач обучения, а не из абстрактных «матриц возможностей».

И последнее (отнюдь не по своей важности) замечание. Для того чтобы экспертиза не превращалась в фикцию (а примеры такого рода приводятся практически на каждой учительской конференции), необходимо привлекать для ее проведения высококвалифицированных и соответствующим образом оплачиваемых экспертов.

Решение семинара-совещания

Всероссийского семинара-совещания проректоров и заведующих кафедрами информатики педагогических вузов Российской Федерации по проблеме информатизации педагогического образования (19—21.05.92 г., г. Омск).

В работе семинара-совещания приняли участие представители 29 педагогических вузов России, а также ряда педагогических университетов и институтов стран СНГ (Казахстан, Узбекистан, Беларусь). Совещание проходило в обстановке конструктивного обсуждения конкретных предложений по основным проблемам информатизации в системе подготовки педагогических кадров. В основу обсуждения был положен обширный пакет организационно-методических и учебных материалов, подготовленных оргкомитетом на базе республиканского Центра новых информационных технологий в образовании при Омском педагогическом институте с участием других педагогических вузов России. Всего в пленарных заседаниях и рабочих группах в ходе совещания приняло участие более 60 человек.

В ходе семинара-совещания были обсуждены: текущее состояние подготовки учителей в области информатики и НИТО;

стандарт требований к подготовке будущих учителей в области информатики и информационных технологий обучения, предложенный Омским педагогическим институтом;

организационно-методические и учебные материалы для подготовки учителей математики и информатики, учителей естественнонаучного и гуманитарного циклов;

вопросы технического обеспечения информатизации в подготовке педагогических кадров, состоянии работ по компьютерным коммуникациям и АСУ в сфере образования.

Особое внимание было уделено рассмотрению «Основных направлений развития информатизации образования», утвержденных Министерством образования России, и выработке путей их реализации.

Обсуждение показало, что:

в последние годы педагогическими вузами России накоплен определенный опыт по подготовке учителей для преподавания информатики в общеобразовательной школе, по включению элементов информатики и информационных технологий обучения в подготовку педагогических кадров независимо от профиля их специализации; в ряде высших и средних педагогических учебных заведений ведется активный поиск по уточнению объемов и содержания подготовки с учетом перспектив перехода на многоступенчатую структуру

образования, в то же время во многих из них до сих пор используются устаревшие учебные планы и программы, учебные пособия, которые не соответствуют современным требованиям;

серьезным препятствием для решения вопросов компьютерной подготовки учителей является отсутствие доступной по ценам и достаточно эффективной, совместимой и надежной вычислительной техники, а также системы ее обслуживания в учебных заведениях;

отсутствие доступных для отрасли просвещения глобальных сетей связи, предусматривающих возможность доступа в мировые сети, препятствует реализации современных подходов к обеспечению информационных коммуникаций в сфере образования;

правительство России, Министерство науки, высшей школы и технической политики, Министерство образования Российской Федерации (МО РФ) не обеспечивают требуемого уровня финансирования исследовательских работ по компьютеризации в области подготовки педагогических кадров; это обстоятельство сдерживает внедрение актуальных научно-исследовательских проектов (НИП) для сферы образования;

сфера образования и подготовки кадров не имеет достаточного количества периодических изданий по проблемам компьютеризации;

в структуре МО РФ отсутствуют эффективные органы, координирующие работы в области информатизации образования.

Участники семинара-совещания констатируют, что обсужденная программа МО РФ основных направлений информатизации образования имеет конструктивный характер, а намеченные пути ее реализации позволяют надеяться на позитивные сдвиги уже в ближайшее время.

Участники семинара-совещания считают, что:

1. Министерству образования РФ необходимо срочно решить вопрос о создании организационно-методических структур, координирующих работы по информатизации в сфере подготовки педагогических кадров; учебно-методического объединения, научно-методических советов (комиссий); целесообразно создать УМО по информатизации образования МО РФ на базе Омского педагогического института.

2. Необходимо срочно подготовить и представить в МО РФ следующие первоочередные организационно-методические документы и материалы: модель учебного плана педагогического вуза, предусматривающего стандарт требований к компьютерной и технологической подготовке бакалав-

ра, дипломированного учителя, магистра;

учебный план специальных отделений (факультетов) для профильной подготовки бакалавра наук и образования по информатике как основной специальности;

программу работ по разработке и изданию (тиражированию) учебно-методического обеспечения подготовки учителей всех специальностей в области информатики и НИТО (учебные программы, учебные пособия, методические материалы, пакеты ПС и т. п.);

программы и предложения по организации повышения квалификации преподавательских кадров педагогических вузов.

3. При определении государственного стандарта требований к содержанию подготовки учителей в области информатики и информационных технологий обучения за основу могут быть приняты предложения Омского педагогического института.

В то же время каждый педагогический вуз вправе создавать и реализовывать собственные учебные планы.

4. Следует поддержать реализацию экспериментально опробованных технических предложений по организации компьютерных телекоммуникаций в интересах системы образования: построение системы на основе модемов РСМ (предложено Тульским педагогическим институтом); создание распределенной сети на основе уплотнения существующих ТВ каналов связи по проекту РОСЦИО и компании «Панорама»; построение четырехуровневой среды связи для информационной сети системы информатизации народного образования (проект ЕСИНО, предложенный МЭИ и ОКБ МЭИ).

5. Министерству науки и Министерству обра-

зования РФ рассмотреть вопрос о создании периодических журналов по проблемам информатизации в системе образования и подготовки кадров. При этом целесообразно иметь отдельное периодическое издание по проблемам подготовки учителей в педвузах.

Участники семинара-совещания особо подчеркивают, что для успешного развития информатизации в образовании Министерству образования России необходимы:

программа научных исследований по вопросам информатизации образования, отражающая основные направления, рассмотренные на совещании; программа обеспечения высших и средних образовательных учреждений средствами вычислительной и организационной техники.

Главным управлением педагогического образования, развития и координации научных исследований МО РФ нужно принять срочные меры для получения необходимых объемов финансирования для реализации указанных программ на 1992 и последующие годы.

Участники семинара-совещания выражают благодарность Омскому педагогическому институту, республиканскому Центру НИТО за хорошую организацию работы совещания, обеспечение пакетом рабочих документов и материалов.

Участники семинара-совещания рекомендуют всем педагогическим вузам России, не принявшим участие в работе семинара-совещания, приобрести подготовленный оргкомитетом пакет документов и материалов, содержащий актуальные предложения по компьютеризации подготовки педагогических кадров при переходе к многоуровневой структуре образования.

Конкурс ВТ

В Министерстве образования РФ был проведен конкурс производителей вычислительной техники для учебных компьютерных классов. Конкурс проводился в два тура. На первом участники представляли документацию в соответствии с приводимыми ниже требованиями. Прошедшие на второй тур принимали участие в конкурсном натурном показе вычислительной техники.

Технические требования к представляемой ВТ. Программно-аппаратная совместимость с ПЭВМ типа IBM PC XT/AT. Обеспечение соблюдения требований безопасности (электрической, механической, пожарной, электромагнитной), санитарно-гигиенических норм и правил в соответствии с действующими на территории России стандартами, нормами.

Требования к документации. Предоставляется техническое описание аппаратных средств, в котором должны быть приведены:

1. Поблочный состав представляемого комплекта техники.

2. Характеристики используемой элементной базы.

3. Надежностные параметры комплекса ВТ.

4. Техническое описание системного блока с указанием типа процессора, основных параметров процессора; наличия сопроцессора либо возможности его подключения; объема ОЗУ, возможности наращивания; слотов расширения (число, тип); числа и типов портов ввода/вывода; типов дисководов и их характеристик; требований по электропитанию; уровня шума вентилятора; габаритных размеров; дополнительных средств и возможностей.

5. Техническое описание монитора с указанием цветности; размеров; частоты регенерации изображения; размера точки по полю экрана; характеристик адаптера (количество точек на экран, возможность работы с ТВ видеосигналом).

6. Описание клавиатуры.

7. Описание дополнительных устройств, которые могут быть поставлены (принтер, манипулятор «мышь», локальная вычислительная сеть, проекционный телевизор, приборный интерфейс и комплекс для лабораторных занятий, устройства сопряжения с аудио- и видеотехникой и т. д.).

Предоставляется также инструкция по эксплуа-

тации; схема расположения Э-6.

Документы должны включать описание поставляемых программных средств, включая операционную систему, инструментальное обеспечение, пользовательский интерфейс, прикладное программное обеспечение.

В документации должны быть приведены сведения о производителе ВТ, включая мощности производства (количество комплектов в год); характеристики производственных помещений, определяющие качество выпускаемой продукции, а также ориентировочная стоимость представляемых изделий в действующих ценах.

Решение

экспертной комиссии по итогам II тура конкурса IBM-совместимой вычислительной техники для системы народного образования

Во II туре конкурса учебной вычислительной техники были рассмотрены следующие комплекты:

Таблица 1

Изготовитель	Модель
Фирма IBM Фирма АО «Прогресс», Белгород МПО ВТ, Минск «Счетмаш», Курск	PS/1 «Русич-16» ЕС 1849, МК 88.04 «Искра-1031»

Среди иностранной техники первое место присуждено комплекту учебно-вычислительной техники PS/1, представленному фирмой IBM,

Фонд задач Ассоциации

Ассоциация учителей информатики и жюри Всероссийских олимпиад по информатике создали Фонд задач, основными целями которого являются:

сбор, систематизация и распространение методических материалов (задачи, решения, формы проведения различных мероприятий со школьниками по информатике, программное обеспечение такой деятельности и др.);

практическая помощь в организации и проведении различных этапов соревнований школьников по информатике;

работа с авторами задач по включению их материалов для проведения мероприятий со школьниками.

Ближайшие планы Фонда:

сбор, систематизация, распространение текстов задач прошедших краевых, областных олимпиад школьников по информатике;

подготовка и распространение методических рекомендаций и задач для школьного, районного и областного (краевого) этапов Всероссийской олимпиады школьников по информатике

второе место — комплекту учебной вычислительной техники фирмы UNISYS.

Комиссия рекомендует АО «Прогресс» в процессе подготовки производства КУВТ «Русич-16» обеспечить надежность работы локальной сети, развернуть центры технического обслуживания, обеспечить гибкость комплектации по требованию заказчика.

Комиссия единогласно не рекомендует использование в учебном процессе КУВТ, комплектующихся мониторами класса CGA.

Комиссия обращает внимание на необходимость получения одобрения Государственным комитетом по санитарно-эпидемиологическому надзору при Президенте Российской Федерации на разработки оборудования для учебных заведений.

Проверка проводилась набором программных средств, заведомо работающих на модели IBM PC/XT. Проверялось также соответствие комплексов санитарно-гигиеническим требованиям и нормам.

Среди отечественной техники первое место присуждено комплекту «Русич-16», второе место не присуждено, а остальная техника ранжирована в следующем порядке:

Таблица 2

РМП	РМУ
ЕС 1849 «Искра-1031»	МП 88.04 «Искра-1031»

1992/93 учебного года;

подготовка основных положений о деятельности Фонда.

Предполагаемые формы деятельности Фонда: индивидуальные члены, регулярно поставляющие свои материалы, получают материалы Фонда бесплатно (кроме оплаты за копирование и пересылку);

органы народного образования, использующие материалы Фонда для проведения олимпиад и других мероприятий со школьниками, оплачивают их стоимость;

авторы задач получают плату за представленные материалы в зависимости от их использования на различных уровнях соревнований школьников.

Всем заинтересованным лицам и организациям свои предложения и материалы направлять координатору Фонда — Алексею Александру Владимировичу по адресу: 660049, Красноярск, проспект Мира, 68, ГлавУНО администрации края, комната 402, рабочий телефон: (391-2) 27-96-70.

Центр интерактивных средств обучения

Новые информационные технологии все шире внедряются в систему образования. Сегодня это понятие включает не только компьютеры. Появляются новые средства, расширяется использование традиционных.

Центр интерактивных средств обучения (ЦИСО) Министерства образования РФ открыт в августе этого года, но уже сегодня в ЦИСО имеется

большой выбор учебных видеофильмов, кино- и диафильмов для всех уровней образования по широкому спектру предметов. Адрес Центра интерактивных средств обучения: 125315, Москва, Часовая ул., д. 216, Телефон: (095) 155-87-37.

Г. АБУМОВА,
директор ЦИСО

126

СПРАВОЧНЫЙ ЛИСТОК

Разъясняем нормативные документы

С момента выхода нашего последнего номера произошли изменения в условиях оплаты труда преподавателя ОИиВТ. Мы хотим познакомить наших читателей с выдержкой из постановления Минтруда РФ № 16 от 04.06.92 г. «О размерах надбавок и доплат работников образования».

«Во исполнение указа Президента Российской Федерации от 13 мая 1992 г. № 509 «О дополнительных мерах по социальной защите и стимулированию труда работников образования» Министерство труда и занятости населения Российской Федерации по согласованию с Министерством финансов Российской Федерации и Министерством образования Российской Федерации постановляет:

1. Установить размеры надбавок и доплат в процентном отношении к ставкам и окладам учителей и других педагогических работников (включая руководителей учреждений образования) взамен действующих выплат в абсолютных размерах согласно приложению.

2. Надбавки и доплаты, предусмотренные настоящим постановлением, применяются к ставкам и окладам работников образова-

ния, установлены в соответствии с решениями Президента и Правительства Российской Федерации».

Постановление подписал первый заместитель министра труда и занятости населения Российской Федерации В. Космарский.

В настоящее время в Правительстве Российской Федерации рассматривается новый вариант оплаты учительского труда. О всех изменениях мы сообщим вам в следующих номерах нашего журнала. Следите за нашими публикациями.

Приложение
к постановлению Минтруда России
от 4 июня 1992 г. № 16

Размеры
надбавок и доплат к ставкам и
окладам работников образования

Надбавки и доплаты	В процентах
II. Доплаты	
2.8. Учителям, преподавателям или другим работникам за обслуживающие вычислительной техники за каждый работающий компьютер	5

Dear readers,

The "Informatics and Education" editorial staff much regrets that it has managed to publish and deliver you only 4 issues of the magazine instead of 6 ones planned out for 1992. You are aware of the present difficulties in the economic situation in Russia. Serious problems in publishing and printing industry lead even to result that some well-known newspapers and magazines are being closed. Having considered all the above please accept our sincere apologies for this annoying fact. Now we are committed to do everything possible to provide you with the complete set of 6 issues for 1993, especially as "Informatics and Education" is the only scientific-methodic magazine in Russia completely

devoted to introducing and using of New Information Technologies in educational systems.

You are welcome to cooperate with our magazine and we pledge to devote our resources to promoting our mutually beneficial partnership.

We would be proud to have you among the subscribers of "Informatics and Education" for the next year.

Address: Vavilova str., 37, Moscow, 117312, Russia.

Phone (7-095) 124-80-62 Fax: (7-095) 1254380.

Editor-in-chief
V. Melnicov

127

ДОРОГИЕ ЧИТАТЕЛИ!

**Редколлегия и редакция поздравляют Вас
с наступающим Новым годом,
желают здоровья, счастья, больших творческих успехов!
Уверены, что в 1993 году
журнал «Информатика и образование»
будет Вам полезен в педагогической работе,
знакомстве с новыми информационными технологиями.
Ждем от Вас интересных материалов, писем и предложений.**



ЭКСИТОН

Официальный дилер завода "ЭКСИТОН" -
фирма "Колледж" - предлагает:

- компьютерный класс на базе БК-0011М (УК-ИЦ 01М)
- БК-0010.01, БК-0011М
- комплектующие для БК : блоки питания, контроллеры, дисководы и т.д.

Комплект поставки для учебных заведений:

- РМП в составе:
 - БК-0011М
 - печатающее устройство
 - дисковод
 - контроллер дисковода
- РМУ в составе:
 - БК-0011М (от 2 до 12)
- пакет учебных и прикладных программ по курсу "ОИ и ВТ"

Комплект поставки для индивидуальных пользователей:

- Рабочее место в составе:
 - БК-0011М
 - дисковод
 - контроллер дисковода
 - печатающее устройство

К КОЛЛЕДЖ

По вопросам поставки обращаться по адресу:
107005, Москва, Волховский пер., д.11, фирма "Колледж"
телефон 267-70-58

"Завуч"

Составление
расписания
занятий

Система построения расписания занятий для школ и средних специальных учебных заведений. Позволяет автоматизировать весь процесс составления расписания.

"Завуч" работает на ДВК, УК-НЦ, БК-001М, КУВТ-86, IBM.



Интерпретатор
Контролирующих
Систем

Позволит создавать контролирующие программы и опросы по любым учебным предметам. В тексте, выводимом на экран, могут использоваться до 32 псевдографических символов.

ИКС работает на БК-0010, БК-0011, БК-001М, КУВТ-86, УК-НЦ 01.01.



КОЛЛЕДЖ

• Поставка комплектов столов для учебных компьютерных классов (12РМУ + 1РМП)

• Техническое обслуживание классов

"Эскорт"

Обучение
работе
на IBM PC

Система обучения работе на IBM-совместимых персональных компьютерах.

- устройство персонального компьютера
- основы операционной системы MS DOS
- оболочка Norton Commander

Базовый пакет учебных программ по информатике

к учебникам В.А. Каймина


- клавиатурный тренажер
- редактор текстов
- графический редактор
- музыкальный редактор
- электронные таблицы
- табличные базы данных
- интерпретатор Пролога
- антивирусная защита

Разработка

Наш программно-техническое
адрес: обеспечение
107005 по заказу
Москва,
Волховский
пер., д. 11.
фирма «Колледж»
тел. 267-70-58

Цена 4 руб.
70423

155/073

ИНФ 
5-6 1992 **О**

**ИНФОРМАТИКА
И ОБРАЗОВАНИЕ**

