

ИНФОРМАТИКА И ОБРАЗОВАНИЕ

4 1993



НПЦ «САПСАН» и «Дискавери — Информатика» развернули производство и продажу современных компьютерных классов типа «Абакус» из импортных комплектующих на базе IBM PC-совместимых 286/386 компьютеров. При сборке этой техники используется прецизионное западное оборудование контроля качества продукции, не имеющее аналогов в России.

Компьютеры «Абакус» полностью соответствуют западным стандартам качества, предъявляемым к технике, рекомендованной для учреждений образования.

Для обслуживания компьютерных классов типа «Абакус» создана сеть распространения, технической и программной поддержки, обучения пользователей по большинству регионов России и странам ближнего зарубежья.

За подробной информацией и по вопросам поставки техники обращайтесь по телефону: 277-95-67



**Автомат для установки микросхем «Amistar»
На 1-й стороне обложки фрагмент компьютерного класса типа «Абакус».**

ИНФОРМАТИКА И ОБРАЗОВАНИЕ

ИНФ
4'93

Научно-методический журнал
Учрежден Министерством
образования РФ
и коллективом редакции

Издается с августа 1986 г.
Выходит шесть раз в год

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

БОЛОТОВ В. А.
БРЕДИХИН Г. А.
ВАСИЛЬЕВ Б. М.
(зам. гл. ред.)
ЗАЙДЕЛЬМАН Я.Н.
ЗУБЧЕНКО А. А.
КИСЕЛЕВ Б. Г.
КОРОЛЕВ В. А.
КРАВЦОВА А. Ю.
(первый зам. гл. ред.)
КРАСНОВ А. Я.
КУЗЯКИН А. П.
КУРНЕСHOVA Л. Е.
ЛАПЧИК М. П.
ПАХОМОВА Н. Ю.
САМОВОЛЬНОВА Л. Е.
САПРЫКИН В. И.
СМЕКАЛИН Д.О.
УВАРОВ А. Ю.
УТРИНОВИЧ Н. Д.
ФУРСЕНКО А. И.
ХОРОШИЛОВ В. О.
ХРИСТОЧЕВСКИЙ С. А.
ЧУРИКОВ П. А.

СОДЕРЖАНИЕ

ОБЩИЕ ВОПРОСЫ

Конкурс
Смекалин Д.О. Компьютеры для школы 3
9

МЕТОДИКА

Зайдельман Я.Н., Самовольнова Л.Е., Лебедев Г.В. Три
кита школьной информатики (окончание) 13
Сливина Н., Чубров Е. Компьютер на уроках математики 18
Сафьянников А.И. Забавный компьютер 24

НОВЫЕ ПРОЕКТЫ

Моисеева М.В. О компьютерных телекоммуникациях 31
Гуткин М.Л. Информатизация образования и
компьютеры Macintosh 39

ПЕДАГОГИЧЕСКИЙ ОПЫТ

Карпов М.Д. Универсальная тест-программа для
проверки знаний в школе 43
Юдина А.Г. Бейсик или Паскаль? Лого, конечно, лучше! 51

ТОЧКА ЗРЕНИЯ

Пиликов Н.П., Волотин Б.Н., Казудин Н.Т.,
Соловьев Д.Э. Работать на компьютере научит компьютер! 65

РЕДАКЦИЯ:

Отдел методики
Луцкая Н.А.
Отдел ВТ
Усенков Д.Ю.
Отдел рекламы
Васильева Н.А.
Отдел подписки и распространения
Виленкина Е.Н.
Художественный редактор
Иванова Т.В.
Экономический отдел
Бородаева З.В.
Технический редактор
Луговская Т.В.
Корректор
Богданова И.В.

ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР

Шульман К.А. Компьютеры IBM PS/1 — в России	75
Зимарев А.В., Данилин А.А. Система подготовки контролирующих программ Auto Control	77
Архивирующие программы и работа с ними	79
Агеенко Е.И. Архиватор LHA (Версия 2.11)	81
Платонов Н.Е. CD ROM технологии	88
Приглашение в СТЕКЕР	90

КЛУБ УКНЦ

Брусенцев В.А. Винчестер для УКНЦ	93
Штарёв М. Е. Программа исследования функций одной переменной	94

КЛУБ «КОРВЕТ»

Степанов В.Р., Кремчуков Е.Н. Использование спрайтов в Бейсик-программах на «Корвете»	101
Компьютерная графика	106
Нужа В.В. ЛинТех представляет графический редактор «Акварель»	109
Никитин А. LUT или «ячейка FAFB»?	114

КЛУБ БК

Усенков Д.Ю. Реализация многоцветной закрашки на Бейсике (по следам одной старой ошибки)	115
Сапегин И.А. О приручении клавиши «Стоп»...	124
Аскеров Р. Если нет принтера...	125

КЛУБ «АГАТ»

Степанов М.Е. Звездочки, дуги и внутренний голос	127
--	-----

Почту направлять по адресу: 103051, Москва, ул. Садовая-Сухаревская, д. 16, к. 9, журнал «Информатика и образование»

Телефон (095) 208-30-78
 Факс (095) 208-67-37

При перепечатке материалов ссылка на журнал обязательна.

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

Подписано в печать с оригинал-макета 29.11.93. Формат 70x100 1/16. Бумага офсетная. Печать офсетная.
 Усл.печ.л. 10,40. Усл.кр.-отг. 11,70. Уч.-изд. л. 13,5. Тираж 22 700 экз. Заказ 1679

ОБЩИЕ ВОПРОСЫ

КОНКУРС

*Министерство образования Российской Федерации
объявляет конкурс
на участие в реализации Государственной научно-технической программы
«Развитие образования в России»
(направление III «Информатизация образования») на 1994 год*

Основные работы по направлению III «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 по базовым курсам основ информатики.
- 1.2.3. Учебно-методические комплексы профильных и факультативных курсов информатики.
- 1.2.4. Учебно-методические комплексы компьютерно-ориентированных курсов для профессиональной подготовки учащихся по специальностям, связанным с применением ИТ.
- 1.2.5. Учебно-методические комплексы компьютерно-ориентированных 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.
- 1.3.2. Новые информационные технологии в профессиональной подготовке учащихся.
- 1.3.3. Компьютерно-ориентированные 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 НИТ в работе управленческих кадров системы образования.
- 2.1.2.** Информационное обеспечение управляющих подразделений системы образования.
- 2.2.** Создание общего информационного пространства разработчиков и потребителей НИТ 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 НИТ 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 НИТ, нацеленных на применение в коррекционных образовательных учреждениях.
- 3.1.2.** НИТ 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 НИТ 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 НИТ в обучении.
- 4.2.** Переподготовка педагогических кадров.
- 4.2.1.** Учебно-методические комплексы для переподготовки специалистов к использованию НИТ в своей работе.
- 4.2.2.** Учебно-методические комплексы самостоятельной подготовки педа-

готовов к использованию НИТ 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, связанным с использованием НИТ.
- 4.4.3. Применение НИТ для переподготовки временно нетрудоустроенного взрослого населения.

Принять участие в реализации программы могут как организации, так и отдельные коллективы авторов.

Структура заявки научно-исследовательского проекта на участие в Государственной научно-технической программе «Развитие образования в России» (направление III «Информатизация образования») на 1994 год

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 имеющих в практике образования аналогов).

3. Техничко-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 помещения, 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 1994 год:

- *материально-техническое обеспечение,*

- *кадровое обеспечение,*

- *финансовое обеспечение, в том числе:*

- *фонд заработной платы, фонд на оборудование, иные расходы.*

По результатам конкурса отбираются наиболее перспективные заявки. Авторы (авторские коллективы), представившие эти заявки, получают целевое финансирование (грант) на разработку проекта.

Заявки на участие в конкурсе просьба присылать в четырех экземплярах в срок до 15 января 1994 г. по адресу:

Москва, 101856, Чистопрудный бульвар, 86, Министерство образования Российской Федерации, Управление развития и координации научных исследований.

Тел.: 925-17-16; 924-29-93; 924-87-47.

Далее публикуется перечень заявок научно-исследовательских проектов на 1993 год, отобранных в ходе экспертизы и включенных в Координационный план научных исследований.

Выписка из Приложения 1 к приказу
Министерства образования Российской
Федерации № 243 от 10.06.98 г.

КООРДИНАЦИОННЫЙ ПЛАН НАУЧНЫХ ИССЛЕДОВАНИЙ ПО ГОСУДАРСТВЕННОЙ НАУЧНО-ТЕХНИЧЕСКОЙ ПРОГРАММЕ «РАЗВИТИЕ ОБРАЗОВАНИЯ В РОССИИ»

направление III «ИНФОРМАТИЗАЦИЯ ОБРАЗОВАНИЯ»

(разделы 9-12) на 1993 г.**

I. Информатизация обучения

1. Технология проектирования педагогических программных продуктов для курса раннего обучения информатике (Классификаторы и Конструкторы). (Предприятие «Роботландия», Первин Ю.А.)
2. Учебно-методические комплексы для факультативной и кружковой работы по новым информационным технологиям. (Московский городской экспериментальный центр компьютерного обучения, Ладанов В.И.)
3. Проблемы создания и внедрения новых информационных технологий в экологическом образовании. (Уральский ГПИ, Шолохович В.Ф.)
4. Информационно-методическая телекоммуникационная связь в системе «ГНПБ им. К.Д. Ушинского — школьная медиатека». (Перспективы развития информационной инфраструктуры школы — одно из важнейших условий создания общего информационного пространства в образовании). (ГНПБ им. К.Д. Ушинского, Сизов Б.Н.)
5. Разработка унифицированных инстру-

ментальных средств на основе среды МУЛЬТИМЕДИА для создания обучающих систем и различных приложений. (РосЦИО МО РФ, Рыжов В.А.)

II. Информатизация управления образованием

1. Анализ информационной топографии управления финансами Министерства образования РФ. (Центр маркетинга РОБУТ при Нижегородском ГПИ, Куров И.Е.)
2. Обзор зарубежных периодических изданий, публикующих материалы по использованию информационных технологий в образовании. (ГП «Редакция журнала «Информатика и образование», Фрейман В.Э.)
3. Построение совместной учебной работы педагогов и учащихся с использованием компьютерной коммуникации. (Лаборатория «Телекоммуникация в образовании» научного Совета РАН по комплексной проблеме «Кибернетика», Уваров А.Ю.)
4. Концепция построения Российской образовательной телекоммуникаци-

** Здесь приводится наименование проекта, организация-исполнитель, научный руководитель (Прим.ред.).

онной сети. (Рязанский ГПИ, Гири-венко И.П.)

5. Школьная компьютерная телекоммуникационная сеть SCHOOLNET. (Учебно-методический центр «Информвидео» г. Москва, Инсаров В.В.)
6. Развитие банка данных о потенциале развития системы образования в области информатизации и обеспечение его работы в глобальной сети. (РосЦИО МО РФ, Ваграменко Я.А.)
7. Система сбора, статистической обработки и классификации информации о конкурсных научно-исследовательских проектах. (РосЦИО МО РФ, Запрудский Б.С.)
8. Анализ и прогнозирование развития НИТ в сфере образования. (Институт проблем информатики РАН, Христовичевский С.А.)

III. Информатизация работы коррекционных образовательных учреждений

1. Компьютер в помощь людям с аномалиями в развитии зрения и слуха: азбука Морзе как связующее информационное звено с ЭВМ. (СПбГУ, Марусева И.В.)
2. Компьютерные программы для поддержки начального обучения аномальных

детей («Календарь»). (Центр новых информационных технологий для детей-инвалидов «Инвалтех», Кукушкина О.И.)

IV. Подготовка кадров

1. УМК для информационной подготовки бакалавра образования и учителя основной (базовой) средней школы в предметных областях: «Химия», «Биология», «Русский язык и литература», «Черчение и декоративно-прикладное искусство». (Омский ГПИ, Лапчик М.П.)
2. УМК для подготовки преподавателей информатики (РГПУ им. А.И. Герцена, Румянцев И.А.)
3. Учебно-методический комплекс для обучения педагогов-предметников методу проектирования учебной деятельности с использованием НИТ. (институт проблем информатики РАН, Федосеев А.А.)
4. Игровые мультимедиа-сценарии. (клуб ЮНЕСКО «Интеркультура», Шипилов В.И.)
5. Банк данных УМО педагогических учебных заведений МО РФ по информационной подготовке педагогических кадров (БД УМО ИПО). (Омский ГПИ, Лапчик М.П.)

Д.О.Смекалин,

начальник отдела информатизации Министерства образования РФ

КОМПЬЮТЕРЫ ДЛЯ ШКОЛЫ

В настоящее время IBM-совместимая техника производится и распространяется многими фирмами и организациями, вот почему при ее приобретении следует руководствоваться следующими критериями.

1. Компьютерный класс должен состоять из компьютеров учащихся (обычно 10 шт.) и компьютера преподавателя. Все компьютеры должны быть объединены локальной сетью. Наибольшее распространение получили сети типа Arcnet, Novell.

2. Компьютеры должны быть оснащены процессором типа Intel серии 286 или 386 (ЭВМ с 486-м процессором лучше, но стоят существенно дороже). Для решения учебных задач сопроцессор не обязателен. Рекомендуемая тактовая частота (определяющая скорость работы процессора) не ниже 16 МГц.

3. Для компьютеров учащихся рекомендуется объем памяти (ОЗУ) порядка 1 Мб, для компьютера преподавателя — не менее —4 Мб, что позволит впоследствии расширять его возможности при использовании в качестве издательской системы или системы мультимедиа.

4. Компьютер преподавателя должен быть оснащен «твердым» диском («винчестером») объемом не менее 40 Мб и двумя дисководами для гибких магнитных дисков диаметром 3,5" и 5,25". Желательно наличие одного дисковода и на компьютере учащегося (отметим, что гибкие диски 5,25" значительно дешевле дисков 3,5"), но при этом учителю придется регулярно проверять свой «винчестер» на отсутствие «вирусов». Для этой цели может быть ис-

пользована программа AIDSTEST, желательно последней версии (очередная версия этой программы, подготовленная в ноябре 1993 г., распознавала и уничтожала «вирусы» 696 типов).

5. Для компьютеров учащихся и преподавателя рекомендуется использовать мониторы типа VGA (Video Grafic Array) и SVGA (Super VGA). Санитарно-гигиенические нормы работы с учебными ЭВМ в настоящее время уточняются РАО, но предварительные результаты показывают, что 2–3 часа занятий старшеклассников с компьютерами, оснащенными мониторами SVGA, не наносят вреда их здоровью.

Возможно использование защитных экранов, но для мониторов этого типа оно не является обязательным. Использование стеклянных защитных экранов в принципе улучшает качество мониторов, но требует тщательной установки освещения, так как возможно появление бликов, вредно влияющих на зрение. Эффект же от сетчатых защитных экранов требует дополнительного изучения, так как они снижают не только уровень излучения мониторов, но и качество изображения.

6. Принтер должен быть русифицирован (иметь соответствующее ПЗУ), в противном случае печатать тексты можно будет только с помощью программ — текстовых редакторов, генерирующих шрифты (например, в редакторе «Лексикон» — это графический шрифт, печать «загружаемым» шрифтом будет невозможна).

7. В случае подключения к компьютеру модема для «электронной почты» следует

проверить, поддерживает ли он стандартные протоколы передачи данных (тип V22, V32, V42). Если нет, то использование существующих в Российской Федерации и в мире телекоммуникационных сетей, в том числе и создаваемой в настоящее время образовательной телекоммуникационной сети, окажется невозможным.

8. Приобретаемые компьютеры должны быть не только программно, но и аппаратно совместимы с компьютерами типа IBM, в противном случае могут возникнуть большие сложности с подключением к ЭВМ дополнительных устройств, а также с ремонтом. Можно отметить, что компьютеры типа PS, выпускаемые самой фирмой IBM, аппаратно не совместимы с распространенной в Российской Федерации техникой, поэтому приобретение плат и устройств для замены вышедших из строя может быть осуществлено только в самой фирме или у ее дилеров. Стоимость же запасных частей к этим компьютерам намного превышает стоимость аналогичных по функциям устройств других фирм. В частности, стоимость одной «материнской» платы превосходит стоимость нового компьютера, а стоимость «винчестера» примерно равна ему.

Рекомендуется обратить внимание на следующие характеристики компьютера (приводятся чисто внешние признаки):

а) «материнская» плата должна иметь не менее пяти 16-разрядных «слотов» (разъемов) для подключения дополнительных плат (в последних моделях ком-

пьютеров все устройства могут находиться и на одной этой плате, но в этом случае к ней должны быть подключены разъемы для всех внешних устройств);

б) наличие так называемой «универсальной» платы или заменяющих ее плат в составе:

– плата-контроллер (стандартная плата) дисководов (возможно подключение двух «винчестеров» и двух дисководов гибких магнитных дисков),

– плата с двумя последовательными «портами» (один из них крепится дополнительным разъемом) для подключения «мыши», модема, сети, одного параллельного для подключения принтера и одного игрового для «джойстика»;

в) наличие платы-контроллера монитора (визуально у разъема подключения кабеля монитора VGA имеется три ряда контактов, у платы EGA – два ряда);

г) блок питания мощностью не менее 200 Вт;

д) все устройства и платы компьютера должны быть закреплены специальными кронштейнами.

В настоящее время предлагаемые на рынке IBM-совместимые компьютеры собираются, как правило, в Российской Федерации. За рубежом стоимость комплектующих для одной ЭВМ при оптовой закупке составляет 600-650 \$ США. Соответственно и стоимость класса в России с учетом налогов и разумной прибыли не должна превышать 7-8 тыс. \$.

Компьютерный центр обучения "Луч" приглашает на курсы:
*руководителей администраций,
руководителей органов народного образования,
преподавателей информатики и преподавателей английского языка
школ, университетов и других учебных заведений,
использующих IBM-совместимые компьютеры.*

Качество обучения соответствует международным стандартам и достигается за счет применения интенсивных методов обучения, постоянного тесного контакта слушателей с преподавателями (учебная группа состоит из 4-5 человек), индивидуальной практической работы на компьютере (по 4 часа ежедневно), обеспечения каждого слушателя уникальной литературой по информатике (в личное пользование).

Для заочного обучения и пополнения Вашей библиотеки литература высылается наложенным платежом. Чтобы получить наши каталоги "КНИГИ и ПРОГРАММЫ - ПОЧТОЙ" и расписание занятий на 2-полугодие 1994 года, отправьте *заполненный* конверт нам по адресу: 125438, Москва, а/я 51

Занятия ведут преподаватели высшей квалификации и специалисты-практики.

Программа занятий согласуется со слушателями.

Продолжительность каждого курса 10 дней.

Стоимость обучения за одного слушателя в 1993 году - 195 тыс. рублей,

в 1994 году - 470 тыс. рублей

(в эту стоимость входят книги и обед во время занятий, оплата проживания отдельно). Место проведения: Академия оборонных отраслей, Москва, Кронштадтский бульвар, 37 б. (проезд от ст. м. "Водный стадион" до ост. "улица Лавочкина").

Тел. (095)-454-00-35 Маслов Александр Николаевич;

тел. (095)-456-06-08 Урядов Вячеслав Михайлович;

тел./факс/модем 454-30-18, факс 454-25-12,

E-mail: cipkarp@tekosm.msk.su

Размещение иногородних в общежитии квартирного типа

(стоимость проживания уточняется ежемесячно).



Начало занятий:

- для преподавателей информатики, только получивших компьютеры, 1 ноября 1993 года;

10 января, 21 февраля, 11 апреля, 1 августа 1994 года;

- для преподавателей информатики со стажем работы на IBM-совместимых компьютерах

15 ноября 1993 года;

24 января, 14 марта, 16 мая, 4 июля и 5 сентября 1994 года;

- для преподавателей, которые хотят использовать IBM-совместимые компьютеры при обучении английскому языку,

29 ноября 1993 года; 7 февраля, 20 июня и 5 сентября 1994 года;

- для руководителей школ и других учебных заведений, оснащенных IBM-совместимыми компьютерами,

13 декабря 1993 года; 21 февраля и 18 июля 1994 года;

- для руководителей и работников аппарата администраций, советов, органов народного образования и социальных служб

15 ноября 1993 года; 28 марта 1993 года;

- для секретарей-референтов

1 ноября 1993 года; 24 января, 11 апреля и 1 августа 1994 года;

- для бухгалтеров учебных заведений

13 декабря 1993 года; 7 февраля, 16 мая 1994 года;

- по техобслуживанию, установке и сборке персональных компьютеров

15 ноября 1993 года; 10 января, 14 марта и 18 июля 1994 года;

- для преподавателей университетов, учебных институтов, ИУУ, ИПК

и других высших учебных заведений и системы повышения квалификации

29 ноября 1993 года; 28 марта и 4 июля 1993 года.

Заявки и запросы программ обучения направлять по адресу:

125438, Москва, а/я 51, КЦО "Луч".

ВНИМАНИЕ!

для образовательных учреждений Российской Федерации
цены на 50% ниже коммерческих!

Предприятие «ИнфоМир»

предлагает следующие программные средства:

Система программирования на основе школьного алгоритмического языка КуМир полностью соответствует школьному учебнику (А.Г.Кушниренко, Г.В.Лебедев, Р.А.Сворень. «Основы информатики и вычислительной техники». — М.: Просвещение, 1990, 1993), а также методическому пособию для учителей (А.В.Авербух, В.Б.Гисин, Я.Н.Зайдельман, Г.В.Лебедев «Изучение основ информатики и вычислительной техники». — М.: Просвещение, 1992).

КуМир — это:

- полная поддержка школьного алгоритмического языка;
- мгновенная диагностика ошибки в момент, когда ошибка допущена и во всех строках программы, имеющих отношение к этой ошибке;
- мгновенный запуск программы на выполнение, возможность пошагового и непрерывного выполнения программы, наглядная визуализация процесса выполнения, средства отладки;
- поставляемые вместе с КуМиром исполнители (пакеты программ): Робот, Чертежник, Гратекс (графический пакет), Вездеход, Редактор и др.;
- поддержка информационных моделей исполнителей;
- работа в локальной сети и многие другие возможности.

МикроМир — это удобная и простая система подготовки и обработки текстов, удобная инструментальная среда для разработчика программного обеспечения, настраиваемая на используемые программистом компиляторы, кроме того МикроМир содержит вычислительные возможности: вычисление одних колонок через другие в таблицах, суммирование по строкам, вычисления по формулам, вычисления в режиме калькулятора и пр.

ЯмахаМир включают в себя КуМир, МикроМир, программу печати, клавиатурный тренажер и пакет «Линейная алгебра».

КорветМир включают в себя КуМир, МикроМир, программу печати, клавиатурный тренажер, графический пакет.

ФортранМир и ПаскальМир — это системы программирования на основе языков программирования Fortran-77 и Pascal. ФортранМир и ПаскальМир производят мгновенную диагностику ошибок по всему тексту программы и выполнение ее без процесса компиляции. ПаскальМир включает в себя исполнители, описанные в учебнике информатики.

МикроМир для УКНЦ бесплатно прилагается при покупке систем КуМир и ФортранМир.

Заявки на приобретение направлять по адресу: 103051, Москва, Садовая-Сухаревская, 16, комн.9. Журнал «Информатика и образование». Отдел рекламы.

« _____ » _____ 199_ г.

ЗАЯВКА НА ПРИОБРЕТЕНИЕ ПРОГРАММНОГО СРЕДСТВА

Наименование	Количество
ЯмахаМир	
КорветМир	
УКНЦ	
КуМир	
ФортранМир	

Наименование	Количество
IBM PC	
КуМир	
МикроМир	
ФортранМир	
ПаскальМир	

Адрес: _____

Бланк можно размножать всеми возможными способами.

Предприятие вышлет в Ваш адрес счет, а после его оплаты — программное обеспечение с подробной документацией и методическими материалами.

Я.Н.Зайдельман, Л.Е. Самовольнова,
Г.В. Лебедев

ТРИ КИТА ШКОЛЬНОЙ ИНФОРМАТИКИ*

Алгоритмический язык в курсе информатики

Алгоритмический язык занимает большое место в нашем учебнике. В связи с этим часто задают два вопроса: во-первых, почему в общеобразовательном курсе так много программирования, и во-вторых, почему для программирования выбран придуманный язык, а не один из распространенных языков программирования.

Ответ на первый вопрос вытекает из целей курса, изложенных в предыдущей публикации. Провозгласив развитие алгоритмического мышления в качестве цели, мы нуждаемся в средствах для ее достижения. Этим средством в курсе становится программирование, которое рассматривается не как ремесло управления компьютерами, а как наука о построении алгоритмов.

Ответу на второй вопрос — почему понадобилось придумывать новый язык и как он связан с уже существующими — посвящена настоящая статья.

Языки программирования

В современной информатике можно выделить три основные ветви развития

языков программирования: процедурная традиция, функциональная и логическая.

В процедурных языках программа описывает действия, которые необходимо выполнить. Основные средства, применяемые в этих языках, — величины, в том числе табличные, присваивание, циклы, процедуры. К этой традиции относятся такие языки, как Алгол, Фортран, Бейсик, Паскаль, Си, Ада и многие другие.

При построении процедурной программы необходимо ясно представлять, какие действия и в какой последовательности будут производиться при ее выполнении. Следовательно, процедурное программирование основано на алгоритмическом мышлении и может служить средством его развития.

В функциональных языках программа описывает вычисление некоторой функции. Обычно эта функция задается как композиция других, более простых, те в свою очередь разлагаются на еще более простые и т. д. Один из основных инструментов в функциональных языках — рекурсия, т. е. вычисление значения функции через значение этой же функции от других аргументов. Присваивания и циклов в классических функциональных языках нет.

* Окончание. Начало см.: Информатика и образование. 1993, № 3.

Наиболее распространенные среди функциональных языков Лисп и Рефал. Лисп — язык обработки списков давно и активно применяется в системах искусственного интеллекта. Рефал, построенный на алгоритмах Маркова, удобен для обработки текстов. Заметим, что Рефал единственный из распространенных в мире языков программирования, придуманный в нашей стране.

Промежуточное положение занимает язык Лого, который содержит средства и процедурного и функционального программирования. На начальном уровне, которым почти всегда начинается и часто заканчивается его изучение, он похож на классический процедурный язык. Однако при решении сложных задач обработки данных на первый план выходят функциональные методы. Фактически, Лого — это Лисп, адаптированный для обучения. Заметим, что эта адаптация потребовала включения в язык процедур, переменных, присваивания, т.е. средств, характерных скорее для процедурных, чем для функциональных языков.

Функциональная программа, как и процедурная, описывает действия, которые надо совершить для достижения результата (вызов функции — это тоже действие), но ее построение требует скорее математического, чем алгоритмического мышления. Алгоритмическая сложность возникает обычно при программировании циклов, а их при функциональном подходе нет. Правда, использование рекурсии может привести к еще более сложным проблемам.

В логической традиции программа вообще не описывает действий. Она задает данные и соотношения между ними. После этого можно задавать вопросы. Машина перебирает известные (заданные в программе) данные и находит ответ на вопрос. Порядок и механизм перебора не описываются в программе, а неявно задаются самим языком.

Классическим языком логического программирования считается Пролог, хотя он и содержит некоторые средства явного управления перебором, т.е. процедурные элементы.

Построение логической программы вообще не требует алгоритмического мышления. Здесь нет динамики, нет описания действий, программа описывает статические отношения объектов, а динамика находится в механизме перебора и скрыта от программиста.

Каково соотношение перечисленных традиций в различных вариантах школьного курса информатики? В большинстве случаев, в том числе в учебнике [1], курсы ориентированы на процедурное программирование. В учебнике [2] подробно рассматривается процедурное программирование и обзорно-логическое. О функциональном программировании говорится в некоторых курсах, построенных на базе Лого.

Наш учебник [3] ориентирован на процедурный подход. Это вытекает из целей курса.

Во-первых, процедурное программирование наиболее связано с алгоритмическим мышлением, развитие которого мы считаем основной целью курса.

Во-вторых, курс должен соответствовать науке, а процедурная традиция является сегодня ведущей и наиболее разработанной, несмотря на громкие фразы типа «Пролог — язык программирования будущего» (интересно, видна ли авторам этого лозунга его двусмысленность?)

Процедурные языки

Сделав выбор в пользу процедурной традиции, мы оказываемся перед необходимостью выбора из широкого разнообразия языков.

Процедурные языки можно классифицировать многими способами, но нас интересует, насколько полно в них реализованы понятия, принятые в нашем курсе в качестве основных (см. предыдущую публикацию). Это цикл, вспомогательный алгоритм

(процедура), табличные величины (массивы) и исполнители (общие величины).

Циклы и массивы есть практически во всех языках. Процедуры тоже есть почти во всех языках, кроме Бейсика. Подпрограммы стандартного Бейсика нельзя считать процедурами, так как они не обеспечивают передачи параметров и независимости имен.

С точки зрения реализации исполнителей (в том смысле как они понимаются в главе 3 учебника) языки можно условно разделить на три группы.

Во-первых, есть языки, в которых подобного понятия нет. Это, разумеется, Бейсик и, как это ни грустно, Паскаль.

Для реализации исполнителя необходимо объявить какие-то величины общими для алгоритмов, составляющих тело этого исполнителя, и недоступными для всех остальных алгоритмов.

В стандартном Паскале переменная может быть глобальной, т.е. доступной всем процедурам, или локальной, т.е. известной только в одной процедуре. Можно сделать переменную общей для группы процедур, но в этом случае недоступными для внешних вызовов становятся сами эти процедуры.

Понятно, что язык, в котором понятие не реализовано, делает невозможным полноценное освоение этого понятия, поэтому приходится отказываться от популярных Бейсика и Паскаля.

Во-вторых, есть языки в которых понятие исполнителя не предусмотрено, но его можно реализовать. Это, например, Си, где это делается с помощью многофайловой программы, в которой каждый исполнитель занимает отдельный файл, а его общие величины объявляются статическими, или, как это ни странно, Фортран, где исполнители реализуются с помощью COMMON-блоков.

Пользуясь этими языками, можно программировать исполнителей, но очень трудно этому научиться. (Для сравнения:

освоив цикл «пока», можно смоделировать его даже на Бейсике, но изучая Бейсик, очень трудно освоить цикл «пока».)

Наконец, в-третьих, есть языки, в которые встроены средства реализации исполнителей. Это, например, Симула-67, Смоллток, Ада, Модула-2. Но эти языки слишком богаты и, следовательно, сложны для школьного курса.

Таким образом, возникла необходимость извлечь из существующего многообразия языков основные понятия, несущие общеобразовательное содержание, и сосредоточить их в одном языке, очистив его при этом от ненужных технических подробностей. Двадцать лет назад из подобных соображений возник Паскаль. Сегодня появился алгоритмический язык.

Чего нет в алгоритмическом языке

В алгоритмический язык включены многие тенденции современного программирования, но одной важной вещи в нем нет.

Во всех современных языках, начиная с Паскаля, есть средства конструирования типов. Можно, например, задать тип «день недели», допустимыми значениями которого будут понедельник, вторник, среда и т.д. до воскресенья. Это так называемое перечисление. Есть и другие способы конструирования типов.

Эти средства есть в таких языках, как Ада, Смоллток, Си++, Модула-2. В новых языках уже появляются способы, позволяющие разрабатывать средства конструирования. Приведем такое образное сравнение: можно собирать изделие из готовых деталей, можно иметь набор инструментов для изготовления деталей, а можно иметь станки для изготовления инструментов.

В алгоритмическом языке есть только predetermined типы. Здесь он оказывается в одной группе с устаревшими языками, такими, как Алгол, Фортран, Бейсик.

Несмотря на важность и содержательность конструирования типов, оно оказывается достаточно сложным, и его не удается упростить так, чтобы, сохранив содержание, сделать доступным для общеобразовательного школьного курса. В результате этой ветви не нашлось места в курсе, а следовательно, в алгоритмическом языке.

Как изучать алгоритмический язык?

Основной принцип изучения языка в нашем курсе заключается в том, что язык изучать не надо. Надо решать содержательные, по возможности трудные задачи, по мере усложнения которых возникают потребности в новых языковых средствах.

Эти средства в курсе разбиты на три больших блока.

Первый блок — алгоритмические конструкции (§§4–10 учебника). Это вспомогательные алгоритмы, циклы, ветвления. Этот блок можно назвать «программирование до величин». Наличие этого блока, отработка основных алгоритмических конструкций на наглядных задачах нематематического характера — существенная особенность курса.

Изучать этот блок можно по-разному. Можно, следуя учебнику, разбирать понятия одно за другим, читая параграфы и решая задачи. Можно менять порядок изложения. Можно, особенно в сильных классах, применить метод «шоковой терапии»: разобрать все конструкции сразу, а потом долго решать задачи. Возможны разные методики, важно только, чтобы этот блок был, чтобы алгоритмические конструкции были освоены до величин.

Многие курсы начинаются с понятий величины и присваивания, такой курс можно построить и на основе алгоритмического языка, но у него будет другая идеология, и, несмотря на единство языка, это будет совсем другой курс.

Второй блок — величины (§§11–15). Здесь рассматриваются понятия величины, присваивания, различные типы величин (числовые, логические, табличные, литерные), обмен информацией между алгоритмами.

Этот блок наиболее близок к традиционным курсам, задачи для него можно найти практически в любой книге по программированию.

Третий блок — исполнители (§§21–24). Здесь вводятся общие величины и рассматривается применение полученных знаний для решения больших задач.

Что изучать после алгоритмического языка?

Этот вопрос задают очень часто, особенно сейчас, когда усилилась тенденция переноса курса из 10–11 в 8–9 классы.

По нашему мнению, курс является законченным и не имеет общеобразовательного продолжения. Конечно, курс и учебник можно улучшать, дорабатывать, но он полон по набору понятий, дальнейшее развитие выходит за рамки общеобразовательного уровня и может быть предметом углубленного, факультативного изучения.

Можно выделить несколько направлений подобных факультативов.

Первое, традиционное, направление — языки программирования. Обычно изучают языки, которые реализованы на конкретной доступной технике, разбирают как в этих языках реализованы изученные понятия и начинают создавать реальные программы. Это можно назвать производственным программированием.

С точки зрения развития мышления и освоения новых понятий полезно изучение языков, содержащих какие-то идеи, отсутствующие в алгоритмическом. Это могут быть Паскаль, Си или Си++, Смоллток. Очень полезно рассмотреть Алгол-68. Это язык, на котором не очень удобно программировать, но он богат содержатель-

ными идеями, из которых развились многие современные языки. Интересно познакомиться с непроцедурными языками, это могут быть, например, Лисп и Пролог.

Второе направление – структуры данных. Начиная с простейших паскалевских средств конструирования типов и до сложных структур таких как стеки, деки, очереди, деревья. Материал по этой теме можно найти в учебнике для вузов и во многих книгах, специально посвященных структурам данных.

Третье направление – методы. Речь идет о темах, затронутых в §16 учебника, о построении и доказательстве алгоритмов. Очень точное название этого направ-

ления – «программирование для математиков». Это направление можно рекомендовать для углубленных курсов информатики в математических школах.

Литература

[1] Гейн А. Г. и др. Основы информатики и вычислительной техники. Свердловск: Изд-во Уральского ун-та, 1989.

[2] Каймин В. А. и др. Основы информатики и вычислительной техники. М.: Просвещение, 1989.

[3] Кушниренко А. Г. и др. Основы информатики и вычислительной техники. М.: Просвещение, 1990.

Предприятие «ИнфоМир»

проводит курсы подготовки и повышения квалификации учителей информатики и методистов

В программе 7-дневных курсов: основные понятия информатики и методика их преподавания в средней школе; работа по учебнику А.Г. Кушниренко и др. «Основы информатики и вычислительной техники» с использованием системы КуМир. Новые инструменты организации учебного процесса на ЭВМ — активные гипертекстовые среды. Курсы проводят авторы учебников, методических пособий и программного обеспечения. По окончании курсов выдается сертификат установленного образца, комплект программно-методического обеспечения КуМир, включая обучающие гипертексты. В стоимость входит проживание и заказ обратных билетов.

Для прохождения курсов направляйте заявки по адресу:

125167, Москва, Ленинградский пр-т, 45, корп.2. Тел.: (095) 939-17-86

ЗАЯВКА НА УЧАСТИЕ В КУРСАХ

Наименование организации:

Адрес:

Ф.И.О.:

Телефон:

Факс:

E-mail:

Продолжаем разговор о применении компьютера в обучении математике. В предыдущих статьях (в № 3/4, 1992 г. и № 1, 1993 г.) мы познакомили читателей с некоторыми программами, используемыми на уроках математики, в частности, с программой ФОРМУЛА. В этой статье мы хотим рассказать о нашей методике проведения компьютерных занятий по математике в школе и на младших курсах вуза.

Н. Сливина, Е. Чубров

КОМПЬЮТЕР НА УРОКАХ МАТЕМАТИКИ

Об опыте применения программы ФОРМУЛА

Работа на компьютере не является равновидностью математической деятельности. Некоторые педагоги и исследователи даже считают, что работа на компьютере мешает занятиям математикой. Методики и средства обучения и приобретения математических знаний с помощью компьютера должны стремиться устранить это противоречие.

Широкое распространение компьютерных технологий привело к появлению на рынке большого количества профессиональных пакетов (Mathematica, MatCAD, MatLab и др.), а также сравнительно недорогих, доступных программируемых калькуляторов, предназначенных для проведения математических расчетов. Они обладают широкими графическими возможностями, умеют производить сложные вычисления и выполнять аналитические операции (разложение на множители, интегрирование, дифференцирование, суммирование рядов и т. д.). Попытки непосредственного включения таких средств в учебный процесс привели к необходимости разрабатывать специальные учебные курсы для овладения тем или иным пакетом, создавать систему обучения и переквалификации преподавателей математики. При этом огромное количество усилий и учебного времени затрачивается не на изучение математики, а на освоение специфиче-

ческих технических приемов работы с некоторым вспомогательным инструментом.

Преподаватели математики испытывают множество трудностей при освоении таких средств, неуверенность при проведении занятий и, вследствие этого, распространение методик компьютерных занятий встречает сопротивление с их стороны.

Кроме того, при обучении с помощью таких средств часто происходит подмена одной информации другой – вместо информации о свойствах математического объекта (что предполагалось в задании) учащийся запоминает последовательность действий, обеспечивших получение ответа от компьютера.

Мы согласны, что одной из задач математического образования в наши дни является подготовка к грамотному (в смысле постановки задач, критического анализа и интерпретации результатов) использованию всевозможных прикладных математических пакетов в практической математической деятельности. Однако нам представляется неправильным использование сложных инструментов на начальных стадиях обучения, их место на более высоких уровнях. Мы также полагаем, что обучение – это извлечение информации из окружающей среды и ее накопление для последующего использования.

Наш подход к обучению и получению знаний по математике с помощью компьютера и к построению компьютерных занятий основан именно на таком понимании про-

цесса обучения. Мы стремимся учить извлекать информацию и «повышать ее уровень».

Такой взгляд на процесс обучения (получение информации низкого уровня – повышение уровня информации) определил конструкцию занятий. Сначала сбор отдельных фактов (получение информации низкого уровня), потом выявление закономерностей, описание их, формулировка свойств соответствующего математического объекта (повышение уровня информации). Занятие представляет собой совокупность простых заданий (выполняемых с помощью компьютера) и вопросов к каждому или к серии заданий. Ответы на вопросы учащийся дает самостоятельно, рассматривая на экране множество кривых или чисел. Именно на этом этапе происходит повышение уровня информации, которое в дальнейшем стимулируется контрольными вопросами. Во многих заданиях анализу подвергаются результаты численного или графического эксперимента, проведенного учащимся.

Приведем некоторые примеры.

На компьютерном занятии «Графическое решение систем уравнений», завершающем изучение квадратного трехчлена, в некоторой серии заданий предлагается решить графически системы:

$$\begin{cases} y - x^2 + 2x = 4 \\ y - 2x = 1 \end{cases} \quad \begin{cases} y - x^2 + 2x = 4 \\ y = 2x \end{cases}$$

$$\begin{cases} y = x^2 - 2x + 4 \\ y = 2x - 1 \end{cases} \quad \begin{cases} y = x^2 - 2x + 4 \\ y = 5x + 1 \end{cases}$$

В режиме вычислений учащийся определяет соответствующие функции (рис. 1), строит их графики (рис. 2 и 3).*

Вопросы к заданиям (приведем содержание вопросов, в тексте занятия они

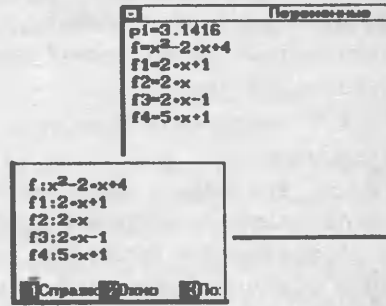


Рис. 1

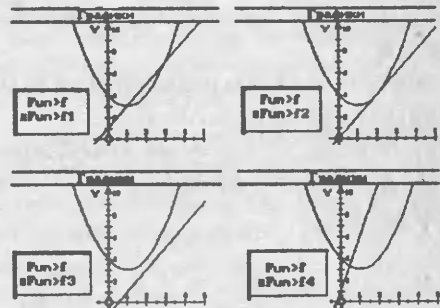


Рис. 2

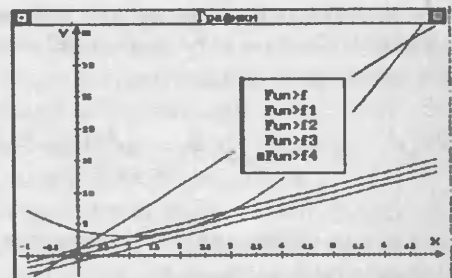


Рис. 3

сформулированы конкретнее и с разумным соблюдением математической строгости):

* Здесь и ниже приведены только существенные фрагменты изображения на экране (подробное описание программы ФОРМУЛА см. в № 1, 1993 г.).

- Пересекаются ли линии?
- Имеет ли система решение?
- Сколько точек пересечения?
- Сколько решений?
- Укажите координаты точек пересечения.

Вопросы, как видно, очень просты, они лишь побуждают внимательно посмотреть на изображение и установить связь количества решений со взаимным расположением линий на графике.

Понятно, что, задав к этой же серии изображений другие вопросы, можно побудить учащегося к анализу других связей, например, разрешимости системы в зависимости от коэффициентов линейной функции.

Задания типа «математический эксперимент» могут быть такими:

а) подобрать значения коэффициентов k и b линейной функции $y = kx + b$ так, чтобы система:

$$\begin{cases} y = x^2 - 2x + 4 \\ y = kx + b \end{cases}$$

имела два решения, одно решение или не имела решений вообще;

б) подобрать коэффициенты квадратного трехчлена $(x - a)^2 + b$ или $ax^2 + bx + c$ так, чтобы система:

$$\begin{cases} y = (x^2 - a)^2 + b \\ y = 5x + 1 \end{cases} \quad \text{или} \quad \begin{cases} y = ax^2 + bx + c \\ y = 5x + 1 \end{cases}$$

имела два решения, одно решение или не имела решений вообще.

Здесь вопросы к заданию, помимо анализа взаимного расположения параболы и прямой, могут вернуть к повторению свойств квадратного трехчлена, линейной функции, правил построения прямой и параболы. Очень полезно решать системы с прямыми вида $x = \text{const}$, $y = \text{const}$. Серией вопросов типа «сравните взаимное расположение» мы побуждаем учащегося выполнить не только графический, но и аналитический анализ системы.

В контрольных вопросах – от простейшего «Сколько решений может иметь система вида...?» до более общего «При каких условиях на коэффициенты уравнений система имеет одно, два, ни одного решения?» – мы побуждаем учащегося сделать выводы в более общей форме, т. е. еще повысить уровень информации. Мы придаем очень большое значение вопросам типа «приведите пример», полагая, что при поиске адекватного примера (особенно в аналитической форме) происходит более глубокое осмысление теории.

Рассмотрим задачу исследования функции с помощью производной. В одном из заданий предлагается построить график функции $y = (x - 1)^2 (x - 4)^3$ и ее производной. Учащийся определяет в режиме вычислений функцию и ее производную (рис. 4), а затем строит разными цветами их графики (рис. 5). Обратите внимание, что тот, кто плохо дифференцирует, тоже справится с заданием. Вопросы к заданию:

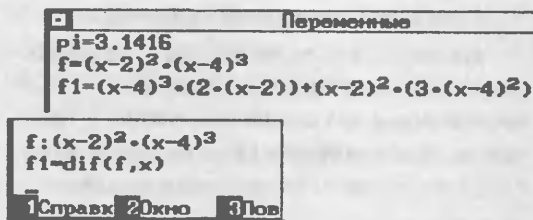


Рис. 4

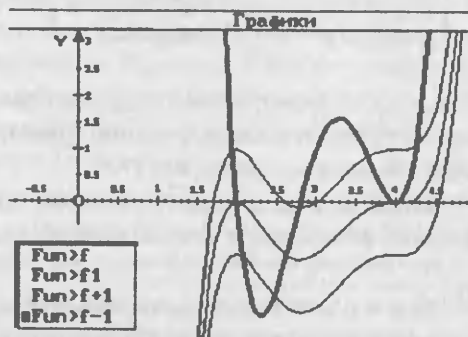


Рис. 5

- Где функция растёт? Убывает?
- Где график производной лежит выше/ниже оси абсцисс?
- Где производная положительна? Отрицательна?
- Имеет ли график максимумы? Минимумы? Укажите их координаты.
- Пересекает ли график производной ось Ox ? В каких точках?
- Имеет ли уравнение $f'(x) = 0$ решения? Укажите их.

В следующей серии заданий более детально исследуется поведение функции в окрестности максимума/минимума, асимптоты графиков, особые точки.

После задания: «Постройте график функции $y = \sqrt[3]{(x+3)x^2}$ и ее производной» (рис. 6) формулируются вопросы типа:

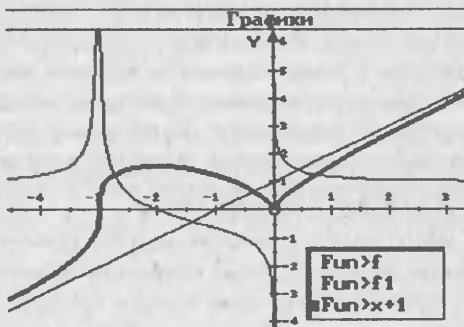


Рис. 6

- Можно ли утверждать, что из равенства нулю производной следует, что в точке есть максимум/минимум? Приведите примеры.
- Можно ли утверждать, что если функция имеет в точке экстремум, то ее производная в этой точке равна нулю?
- Когда справедливо утверждение, сформулированное выше? Приведите примеры.
- Укажите значения пределов производной при стремлении x к нулю слева и справа.

– Имеет ли график производной вертикальную асимптоту? Запишите ее уравнение.

– Как ведет себя функция в окрестности точки $x=0$?

Те же вопросы для точки $x=-3$.

– Укажите значения пределов производной на бесконечности.

– Имеет ли график производной горизонтальную асимптоту? Запишите ее уравнение.

– Есть ли асимптота у графика функции? Постройте на экране прямую $y = x + 1$.

– Можно ли утверждать, что если функция имеет наклонную асимптоту, то ее производная имеет на бесконечности конечный предел? Как найти угловой коэффициент асимптоты?

В очередной серии заданий предлагается по построенному графику функции изобразить схематически (на бумаге) график производной и проверить себя с помощью компьютера. Затем наоборот — по графику производной изобразить схематически график функции. Формулируются вопросы об однозначности решения такой задачи.

В других заданиях предлагается для некоторых из исследованных графически функций построить таблицы значений функции и производной на некотором интервале (в окрестности интересных точек) и дать ответы на те же вопросы.

Контрольные вопросы к такому занятию требуют от учащегося попытки самостоятельно сформулировать те основные теоремы о связи поведения функции и ее производной, которые обсуждались на занятии, привести примеры и контрпримеры (графические и аналитические), построить схематически график производной по графику функции и наоборот.

Приведенные выше примеры не случайно содержат, в основном, анализ графических изображений. Мы считаем, что целенаправленное развитие графической культуры следует начинать как можно раньше. Графическое мышление — эле-

мент образного мышления. Образное мышление доминантно для большинства людей; оно – первично, тогда как логическое – вторично. Развивать логическое мышление следует на базе образного. Графические образы воздействуют непосредственно на органы чувств, их красота доступна не слишком искушенным в математике людям. Многие математические утверждения становятся прозрачнее, понятнее, легче и лучше запоминаются и усваиваются, если они получают графическую интерпретацию. Здесь мы отсылаем читателей к статье И. Шарыгина «Некоторые размышления по поводу школьного курса геометрии» в «Учительской газете» от 9 июня 1992 г.

Как видно из приведенных примеров, на таких занятиях компьютер играет сугубо вспомогательную роль, выполняя за учащегося серию построений и вычислений, т. е. предоставляет ему возможность получить некоторую информацию (играет роль внешнего мира). Извлечение информации производит сам учащийся, отвечая на вопросы к заданиям, таким же образом он повышает уровень информации. Наблюдая графики и числовые множества, учащийся получает самую различную информацию (впечатления?). Задача педагога – так сформулировать вопросы, чтобы обязательно была извлечена нужная в контексте занятия информация и так подобрать последовательность заданий, чтобы учащийся повышал уровень информации.

Таким образом, адекватность обеспечивается педагогом, и только им, путем выбора заданий, последовательностей заданий, вопросов к заданиям, контрольных вопросов, тщательно подобранным домашним заданием. Последнее исключительно важно, так как знания выступают не сами по себе, а проявляются через соответствующие умения, в первую очередь через умение решать задачи. При этом задания подбираются так, чтобы можно было использовать одни и те же занятия для разных уровней программ по

математике, для разных уровней учащихся (например, путем исключения некоторых заданий). Очень важно, чтобы учащийся мог сам адаптировать к себе учебный материал, подчас самостоятельно поднимаясь на более высокий уровень.

Ключевым моментом здесь является то, что методика обучения с помощью компьютера целиком в руках преподавателя, позволяет ему полноценно использовать весь имеющийся в его распоряжении методический багаж, сохранять полезные традиции.

Следующим важным моментом является то, что компьютерные занятия проводятся так и тогда, и только тогда, когда помощь компьютера существенно способствует получению информации и повышению ее уровня. Компьютерные занятия в такой форме легко адаптируются к любым программам, не вступая в противоречие с накопленным методическим и педагогическим опытом, расширяют рамки и способы взаимодействия с учащимися; они не требуют переработки имеющихся учебников и методических пособий, могут опираться на стандартные задачи и формы контроля знаний.

Нам представляется немаловажным, что такая форма компьютерных занятий позволяет использовать самые различные программные и технические средства, лишь бы они обладали соответствующими вычислительными и графическими возможностями и были просты в работе. Однако мы проводим такие занятия, базируясь на программе ФОРМУЛА. Не вдаваясь в детали ее использования, скажем лишь несколько слов в обоснование такого выбора.

Программа разработана как математическая среда для школьников и студентов. Она не универсальна, но имеет достаточно широкие вычислительные, графические и минимально необходимые аналитические возможности.

Освоение программы и работа с ней очень просты. Преподаватель математики или учащийся легко овладевает самостоя-

тельно основными приемами, пользуясь встроенным справочником и экранным меню или 45-минутным вводным уроком.

Запись математических выражений представляет разумный компромисс между общепринятой математической формой и некоторой компьютерной спецификой, которой не нужно специально учиться.

Программу можно использовать очень широко: от 6–7 классов школы до самостоятельной работы в вузе. При этом, переходя на более высокий уровень, учащийся использует возможности программы все шире, как бы «настраивая» инструмент на собственные потребности.

Работа с ФОРМУЛОЙ на протяжении нескольких лет позволяет, используя одни и те же образы, исследовать математические объекты с различных сторон, переходя на более высокие уровни.

Использование одного инструмента в различных учебных дисциплинах будет способствовать «универсализации» знаний по математике именно благодаря использованию компьютера. Например, курс школьной физики содержит достаточно много задач, состоящих из анализа и преобразования математических выражений, много вычислительных задач. Использование ФОРМУЛЫ при решении физических задач «подключит» опыт исследования математических объектов, при-

обретенный при изучении математики. Математические знания начинают использоваться в новом контексте, графическая интерпретация дополняется физической, анализ графических изображений переходит в анализ физических процессов, расширяется диапазон решаемых задач по физике.

Отдельного разговора заслуживает использование ФОРМУЛЫ в курсе информатики или методов вычислений в вузе. Здесь – еще одна сфера ее применения. На ФОРМУЛЕ легко реализовать многие изучаемые в этих курсах численные алгоритмы, не прибегая к программированию. Выполняя алгоритм пошагово, наблюдая промежуточные результаты, имея возможность в любой момент получить графическую иллюстрацию, учащийся лучше понимает механизм вычислений.

К ФОРМУЛЕ прилагаются комплекты методических материалов, насчитывающие к настоящему времени несколько десятков занятий, в которых реализованы описанные выше подходы. Мы приглашаем к обсуждению проблем и сотрудничеству преподавателей математики, физики, информатики. Направляйте ваши письма в редакцию журнала.

Наш адрес: 103051, Россия, Москва, ул. Садово-Сухаревская, д.16, комн. 9а.

Телефоны: (095) 208-30-78 и (095) 151-19-40.

ПРЕДПРИЯТИЕ «КОМПЬЮТЕР-СЕРВИС»

извещает о выходе в свет печатного иллюстрированного издания популярной книги В. В. Перепелкина «Персональный компьютер в школе» (250 стр.)

Начинающий без проблем постигнет азы программирования, бывалый упорядочит свои знания и узнает много интересного, учитель освоит оригинальную методику преподавания машинного варианта школьного курса «Основы информатики и вычислительной техники».

Алгоритмы, блок-схемы, более 100 задач с решениями и 200 задач для самостоятельного решения. У вас не будет проблем в программировании.

Стоимость книги без учета пересылки 0,26 \$ USA. Оплата в рублях по текущему курсу.

Готовится к изданию вторая книга «Лабиринты Ассемблера» (350 стр.). Принимаются предварительные заявки.

Недорого тиражируем программы ПК «Специалист», «Лик», «Синклер».

По России заказы выполняются наложенным платежом. В страны СНГ – предоплата. Заявка + конверт с марками по адресу:

346839, Ростовская область, Неклиновский район, п. Новоприморский.

«КОМПЬЮТЕР-СЕРВИС» ОТДЕЛ МАРКЕТИНГА

А.И. Сафьянников

ЗАБАВНЫЙ КОМПЬЮТЕР

В настоящее время компьютер в начальной школе используется в процессе развития ребенка как источник получения знаний и формирования навыков. Общение с компьютером на уроках способствует развитию у ученика нестандартного мышления, памяти, реакции, сообразительности, коммуникабельности. Получая знания по различным предметам, школьник одновременно обучается начальным правилам работы с ЭВМ. Но использование компьютера в начальной школе ограничено санитарно-гигиеническими нормами (10-15 мин в неделю). Кроме того, компьютерные классы есть не в каждой школе, обучаются в них, как правило, старшие школьники.

Как же научить младшего школьника пользоваться компьютером без больших материальных затрат? Для этого идеально подходит развивающая игрушка американской фирмы «Texas Instruments» — «Computer Fun» (Забавный компьютер). Стоимость ее в 8-10 раз ниже стоимости компьютера, дисплей без вредного излучения.

Эта игрушка воплотила в себе новейшие достижения в области компьютерных технологий и вполне подходит под определение систем «Multimedia»: графический дисплей, средства анимации, звуковое сопровождение. Человеческий голос не только выдается через динамик, но и воспринимается микрофоном.

Эта игрушка стилизована под персональный компьютер типа «Notebook» с удобной ручкой для переноски (масса 1,5 кг). «Computer Fun» — говорящий компьютер. Яркая расцветка, набор интересных развивающих и обучающих игр — все это нравится ребенку и сильно повышает его мотивацию к обучению. В постоянном запоминающем устройстве «Computer Fun» зашиты упражнения по английскому языку, математике, развитию памяти и творческих способностей. В отличие от обыч-

ного компьютера, «Computer Fun» не имеет ни жесткого диска, ни дисководов для гибких дисков. Работать можно только с теми упражнениями, которые хранятся в ПЗУ «Computer Fun».

Компьютер имеет небольшой жидкокристаллический дисплей с возможностью изменения яркости, сенсорную клавиатуру (QWERTY), на которой рисунками обозначены упражнения. Рассмотрим на примере некоторых из них, как работает «Computer Fun».

1. Изучение латинского алфавита.

Инициализируется нажатием на клавишу (abc). При нажатии на любую букву слышим название этой буквы, на экране видим изображение объекта, начинающегося на эту букву, и слышим его название на английском языке.

Например, «d». Слышим «ди», на экране появляется изображение оленя, бьющего копытом, и звучит его название — «Deer». Нажав «d» еще раз, получаем уточку и ее название «Duck».

К каждой букве имеется словарь на 5-7 слов. В этом режиме ученик не только изучает алфавит, но и пополняет свой словарный запас, опираясь на образы. Это чисто обучающий режим.

2. Режим с элементами контроля.

Случайным образом на экране изображается какой-то предмет, ниже — его название, но с пропущенной первой буквой. Это слово также произносится. Ученик должен вставить пропущенную букву. В случае двойной ошибки компьютер подсказывает правильный ответ. Ошибки не подсчитываются.

Например, цветок. Появляется его изображение, произносится «Flower», и на экране пишется «_lower». После этого ученик должен нажать букву «F» и клавишу «Enter». Компьютер произносит «Flower», а также слова поощрения, после чего выдает следующее задание.

3. Этот режим замечателен тем, что

воспринимает ответы ученика, произнесенные в микрофон и так же, как и предыдущий, с элементами контроля.

На экране случайным образом размещены животные, растения, какие-то предметы. В нижней части экрана находится название одного из них. Перемещая стрелками автомобиль, нужно остановить его напротив этого предмета и зафиксировать клавишей «Enter». Компьютер произнесет:

«Соответствует ли рисунок слову? Скажите «Yes» or «No». Нужно произнести в микрофон «Yes». Если все сделано правильно, заиграет мелодия, замигает картинка, будет выдано следующее задание.

4. В этом режиме ученик осваивает составление простых предложений, состоящих из подлежащего, сказуемого и дополнения. Нажимая на одну из управляющих клавиш, можно подбирать слова, подходящие по смыслу. Когда предложение составлено, компьютер произносит его вслух, и на экране появляется изображение того, что описывает предложение.

Например, вы составляете такую фразу: «A lion talks to the frog». Компьютер произносит эту фразу, а на экране демонстрируется короткий мультфильм: лев подходит к лягушке и говорит «Hello», а затем «Good bye».

5. Последний набор упражнений по изучению языка позволяет составлять целые истории, формируя и закрепляя навыки чтения и набора слов с клавиатуры.

Подобным же образом в игровой манере подобраны упражнения по математике и тренировке памяти, которые сопровождаются комментариями на английском языке.

«Computer Fun» может быть использован в начальной школе на уроках иностранного языка (кроме английского, есть еще игрушки, «говорящие» на немецком, французском, итальянском языках), на уроках математики, первых уроках информатики.

Ребенок с радостью будет играть с «маленьким компьютером» не только в школе, но и в детском саду или дома.

«ЭЛТИ-КФДИЦ»

ПРЕДЛАГАЕТ:

1. Учебные компьютерные классы на базе АТ 286/386 для школ и детских садов. В комплект входит программное и методическое обеспечение, обучение специалистов, гарантийное обслуживание. Стоимость от 4000 USD.

2. Региональные телекоммуникационные образовательные сети. Стоимость абонентского пункта от 1200 USD.

3. Оборудование фирмы «Texas Instruments»:

- калькуляторные классы с учебными пособиями. Стоимость от 350 USD;
- графические калькуляторы нового поколения. Стоимость от 130 USD;
- уникальные развивающие игрушки и языковые тренажеры (английский, немецкий, французский, итальянский). Стоимость от 30 USD.

ВОЗМОЖНА ОПЛАТА В РУБЛЯХ ПО КУРСУ ММВБ

тел.: 392-78-18, 392-62-95

факс: 392-82-90

Адрес: 115409, Москва, ул. Москворечье, 31, корп. 2

ДОСКА ОБЪЯВЛЕНИЙ

Обмен ПО для БК-0010.01.
214000, г. Смоленск, а/я 318.

Системные, учебные (математика, иностранные языки, англо-русский словарь 20000 слов), логические (шахматы, память, мышление), игровые программы для Спектрума. Для каталога — Ваш конверт.
630129, Новосибирск, а/я 410.

Программы для компьютеров: БК, УКНЦ, ZX Sinclair, Atari, Commodore, Amiga, IBM PC. Москва, а/я 9.

Владельцам БК-0010.01 предлагаю переписку, обмен программами и помощь в создании заставок. Имею более 1300 программ. Вышлю каталог.
640000, г. Курган, Пушкина, 123, кв. 20. Прохоров А.А.

Извещаем о выходе в свет печатного иллюстрированного издания популярной книги В.В. Перепелкина «Персональный компьютер в школе» (250 стр.). Начинаящий без проблем постигнет азы программирования, бывалый упорядочит свои знания и узнает много интересного, учитель освоит оригинальную методику преподавания машинного варианта школьного курса «Основы информатики и вычислительной техники». Алгоритмы, блок-схемы, более 100 задач с решениями и 200 задач для самостоятельного решения. У вас не будет проблем в программировании. Стоимость книги без учета пересылки 0,26 \$ USA. Оплата в рублях по текущему курсу.

Готовится к изданию вторая книга «Лабиринты Ассемблера» (350 стр.). Принимаются предварительные заявки. Недорого тиражируем программы ПК «Специалист», «Лик», «Синклер». По России заказы выполняются наложенным платежом. В страны СНГ — предоплата. Заявка + конверт с марками по адресу:
346839, Ростовская область, Неклиновский район, п. Новоприморский.
В.В. Перепелкин.

*Это стоит дорого,
но успех — еще дороже!*

Multivision Pro



ВЫ ХОТИТЕ:

- использовать самую современную технологию так, чтобы Ваша лекция запомнилась?
- сопровождать демонстрацию богатым видео и компьютерным материалом?
- управлять вниманием публики, стоя у большого экрана лицом к аудитории?

ТОГДА ВАМ НЕОБХОДИМ:

комплекс **Multivision Pro**

AIST

Advanced Instructional Software Trading AB

РОССИЯ, 125050, МОСКВА, К-50 а/я № 80, АИСТ АВ; ТЕЛ: 095/2296706, ФАКС: 2297653

Современные технологии ведущих фирм мира сегодня *для Вас!* Фирма AIST предлагает **MultiVision Pro**

это профессиональный демонстрационный программно - аппаратный комплекс. Используя самую современную технику Вы получите высокое качество демонстрации. Новейшие технологии позволят Вам создать собственные курсы, методики и наглядные пособия. Сегодня не Вы в плену у техники, а техника и технология на службе у Вас!



CANADA

Фирма Creative Labs, Inc (Канада) представляет Sound Blaster Pro. Благодаря Sound Blaster Pro Ваша демонстрация зазвучит! Вы сможете ввести в Вашу демонстрацию любой звук, музыку, человеческий голос.



U. S. A.

Фирма Intel (U.S.A.) представляет Smart Video Recorder, который позволит Вам переносить видеоизображение с бытового видеомagneтофона на компьютер и обратно.



U. S. A.

Фирма Hewlett Packard (США) представляет
- цветной сканер, который позволит Вам включать в Ваши программы высококачественные иллюстрации;
- лазерный принтер, благодаря которому Ваши документы будут выглядеть так, будто они только что напечатаны в типографии.



U. S. A.

Фирма Proxima (США) представляет лучшую в мире цветную жидкокристаллическую панель Ovation, уникальную сенсорную камеру Cyclops, и лазерную указку. Ovation проецирует компьютерное и видеоизображение без потери цвета. Cyclops и лазерная указка превращают спроецированный кадр в огромный сенсорный экран и позволяют Вам управлять демонстрацией стоя перед ним или сидя вместе с аудиторией!



SWEDEN

Фирма IST AB (Швеция) представляет MultiVision - суперконструктор обучающих программ. Вы легко можете создать компьютерный учебник в любой предметной области, не зная ни одного языка программирования. Сочетание графики, текста, сканированных иллюстраций и мультипликации поможет не просто эффектно подать лекционный материал, но и оживить его. Теперь Вы не тратите время на поиски наглядных пособий, Вы делаете их сами!



GERMANY

Фирма Medium (Германия) представляет кодоскоп высокой надежности. Совершенство оптики и мощные лампы позволят Вам провести демонстрацию даже в незатемненной комнате!



JAPAN

Фирма Matsushita Electric Industrial Co., Ltd (Япония) представляет

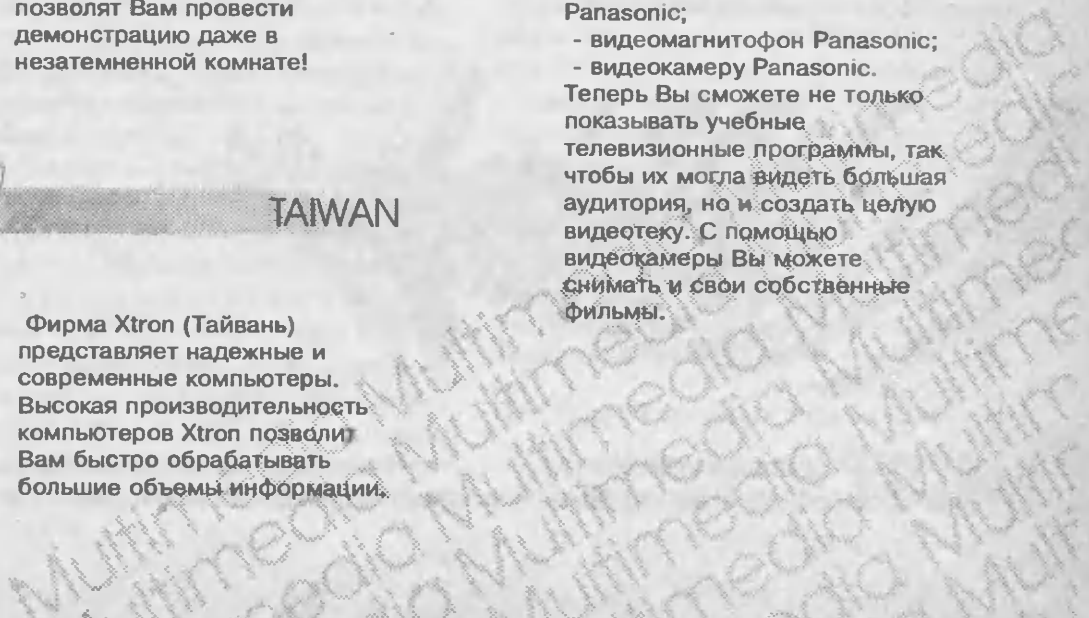
- большой цветной телевизор Panasonic;
- видеомагнитофон Panasonic;
- видеокамеру Panasonic.

Теперь Вы сможете не только показывать учебные телевизионные программы, так чтобы их могла видеть большая аудитория, но и создать целую видеотеку. С помощью видеокамеры Вы можете снимать и свои собственные фильмы.



TAIWAN

Фирма Xtron (Тайвань) представляет надежные и современные компьютеры. Высокая производительность компьютеров Xtron позволит Вам быстро обрабатывать большие объемы информации.



Состав комплекса MultiVision Pro



- Компьютер типа IBM PC/AT-486.
Processor 80486
4 Mb RAM
120 Mb IDE Hard Drive
3.5" 1.44 Mb; 5.25" 1.2 Mb Floppy Drives
Super VGA Card, 1024 Kb
Mini Tower Case + 200 VA Power Supply,
220 V/50 Hz
14-inch Super VGA Monitor 0.28 DPI
3-button Serial Mouse
Keyboard Latin/Cyrillic
- Лазерный принтер HP LaserJet IV.
Разрешение 600 DPI
256 градаций серого цвета
- Цветной сканер HP ScanJet IIc.
Разрешение 400 DPI
16 миллионов цветов
- Плата оцифровки и воспроизведения
звука Sound Blaster Pro
16 - битный стереозвук
Стереокolonки
Микрофон
- Плата оцифровки и сжатия видео-
изображения Smart Video Recorder
Indeo - технология сжатия изображения
PAL, SEKAM, NTSC
- Видеокомплекс Panasonic
многосистемный телевизор,
диагональ 72 см
многосистемный видеомагнитофон
видеокамера
- Кодоскоп (Overhead Projector)
Medium 5000
Световой поток - 5000 Лм
- Цветная жидкокристаллическая
панель Proxima Ovation.
Активная матрица
Разрешение 640*480
2 миллиона цветов
VHS, S-VHS
PAL, SEKAM, NTSC
- Сенсорная камера Proxima Cyclops
Расстояние до экрана - 4 метра
Время реакции - 30 миллисекунд
- Proxima Laser Pointer
Дальность действия - 8 метров
- Конструктор обучающих и демо-
страционных программ MultiVision v 4.2
Интеграция изображений, текста, звука
и анимации
Разработка программ без программиро-
вания
Компиляция для DOS или Windows



НОВЫЕ ПРОЕКТЫ

М.В. Моисеева

О КОМПЬЮТЕРНЫХ ТЕЛЕКОММУНИКАЦИЯХ

Компьютерные телекоммуникации все настойчивее проникают в различные сферы жизни современного общества: в бизнес, финансы, банковское дело, средства массовой информации, науку и образование. Несмотря на то что Россия занимает сейчас 37-е место в мире по уровню телефонизации (являющейся одним из важнейших факторов, обуславливающих развитие компьютерных телекоммуникаций в стране), за последние 2–3 года российские пользователи персональных ЭВМ получили доступ ко множеству зарубежных телекоммуникационных сетей.

На общем фоне развития телекоммуникаций в нашей стране постепенно проявляется и становится заметен процесс внедрения компьютерных телекоммуникаций в сферу народного образования, и, прежде всего, в жизнь современной школы.

Десятки тысяч школ за рубежом и сотни школ в нашей стране за последние 5–7 лет начали использовать возможности современных телекоммуникаций непосредственно в учебной работе. Особенно стремятся участвовать в этом процессе школы из глубинки, где уже есть современная телефонная сеть, но они по-прежнему лишены доступа к современной оперативной информации по различным отраслям знаний.

Некоторые учителя используют телекоммуникации преимущественно для внеклассной работы с учащимися по отдельным экспериментальным проектам. Однако уже сейчас многие школы за рубежом

используют компьютерные телекоммуникации непосредственно на уроках в условиях реального учебного процесса, постепенно подготавливая учащихся к жизни в информационном обществе. Компьютерные телекоммуникации начинают постепенно осознаваться многими педагогами как один из инструментов познания окружающего мира. Причем инструмент этот настолько мощный, что вместе с ним в школу приходят новые формы и методы обучения, новая идеология глобального мышления.

Что же нужно иметь в кабинете информатики для того, чтобы ваша школа также смогла стать абонентом какой-либо действующей телекоммуникационной сети? Как говорится в известном телевизионном рекламном ролике биржевой информационной сети «Телемаркет»: «Вам нужно иметь IBM-совместимый компьютер, модем и... все!» И все? К сожалению, для учителя информатики этого «набора» явно недостаточно, так как потребуются еще пакет коммуникационных программ и учебное программное обеспечение, которое поможет научить учащихся работать в телекоммуникационной среде. А самое важное – учитель должен быть сам хорошо осведомлен о современных компьютерных телекоммуникациях.

Что такое «телекоммуникации»?

Телекоммуникации (греч. tele – вдаль, далеко, лат. communicatio – обще-

ние) — это в широком смысле все средства дистанционной передачи информации, такие, как радио, телевидение, телефон, телеграф, телетайп, телекс, телефакс, а также появившиеся сравнительно недавно компьютерные телекоммуникации.

Компьютерные телекоммуникации считаются сейчас не только самым новым, но и самым перспективным видом телекоммуникаций. Компьютерные телекоммуникации (или просто «телекоммуникации» в узком значении этого слова) — это средства дистанционной передачи данных с одного компьютера на другой(ие) при помощи модемов и телефонной сети.

Основа компьютерных телекоммуникаций

В основе компьютерных телекоммуникаций находятся три основных элемента: компьютер, модем и телефонная сеть.

Персональный компьютер со всеми периферийными устройствами играет в этой связке традиционную роль мощной вычислительной системы, на которой можно выполнять различные виды работ: обрабатывать и хранить разнообразную текстовую и графическую информацию, полученную по компьютерным телекоммуникациям или подготовленную к отправке, распечатывать ее на бумаге, а также управлять процессом обмена данных с удаленным компьютером с помощью специальных телекоммуникационных программ.

Телефонные сети предоставляют возможность передавать данные по коммутируемым либо по некоммутируемым каналам связи. В чем их отличие?

Коммутируемые каналы общего пользования — это обычные телефонные линии, соединяющие в автоматическом режиме телефонных абонентов и использующие при этом коммутационное оборудование релейных или электронных АТС. В этом случае абонент может столкнуться с рядом проблем при наборе номера (вызове другого абонента), так как свободных

линий в групповом оборудовании АТС может в данный момент не оказаться. Коммутируемые каналы экономически эффективны для тех пользователей, у которых объем передаваемой информации небольшой и нет высокой срочности в доставке информации.

Поэтому в тех случаях, когда возникает необходимость срочно передать большие объемы информации, причем строго конфиденциальной, используются некоммутируемые (арендованные или выделенные) каналы, обеспечивающие большую надежность передачи данных.

Передавать данные напрямую из одного компьютера по телефонным линиям на другой компьютер нельзя, так как компьютер использует цифровые сигналы, а телефонные линии — аналоговые. Преобразование цифровых сигналов в аналоговые называется модуляцией, а обратный процесс — демодуляцией. Выполняет такое преобразование специальное устройство — модем (англ. modulate/demodulate).

Модем можно сравнить с профессионалом-переводчиком, который не только осуществляет перевод определенного набора слов с одного языка на другой, но и при необходимости корректирует слова, а также употребляет при переводе более сжатые и краткие фразы, для того чтобы собеседники быстрее и четче поняли смысл сказанного.

Помимо своего основного назначения — модуляции и демодуляции сигналов, — модем выполняет многие другие функциональные обязанности. Например, он может автоматически дозваниваться до абонента, отвечать на телефонный звонок или сообщать о состоянии телефонной линии в момент связи. Все эти функции модема должны контролироваться компьютером, а модем должен сообщать ему о том, что происходит на линии связи в данный мо-

мент, например, какие неисправности системы он обнаружил.

Телекоммуникационная сеть

При объединении нескольких коммуникационных систем образуется телекоммуникационная компьютерная сеть. Большая часть компьютеров, подключенных к такой сети, будет выполнять функции абонентских пунктов.

Абонентский пункт — это оборудованное рабочее место пользователя, который, имея компьютер, необходимый набор периферийного оборудования, модем и телефон, может в удобное для него время подключиться через модем и телефонную сеть к какой-либо телекоммуникационной сети и получает (передает, просматривает в интерактивном режиме) интересующую его информацию.

Для того чтобы все отдельные компьютерные коммуникационные системы составляли единое целое, а информация по сети передавалась круглосуточно, в компьютерной телекоммуникационной сети существуют компьютерные узлы связи, которые еще называют центральными компьютерами сети или хост-компьютерами (англ. host — хозяин). Хост-компьютеры с модемами в отличие от абонентских пунктов постоянно подключены к телефонной сети, и через них связываются между собой все пользователи (абоненты) сети. Все компьютерные узлы связи, работающие в сети, связаны между собой.

В настоящее время в мире действуют сотни различных телекоммуникационных компьютерных сетей. Они создаются для пользователей, которые обладают определенной профессиональной ориентацией или же просто потребностями в доступе к информационным ресурсам. С этой точки зрения телекоммуникационные сети можно классифицировать по группам персональных интересов пользователей: банковские, коммерческие, информационные, специализированные, образовательные и т. п.

Большая часть телекоммуникационных сетей, созданных за последние годы в нашей стране и за рубежом, это, как правило, малые компьютерные сети, или BBS (Bulletin Board System), которые имеют всего один центральный узел — хост-компьютер. Каждая из таких BBS редко насчитывает более тысячи пользователей, и чаще всего она недоступна пользователям из-за рубежа. Точно так же и наши, отечественные, пользователи испытывают сложности при попытке соединиться напрямую с BBS других стран.

Следующим, более «серьезным», типом сетей являются большие (глобальные) телекоммуникационные сети — мощные сети, объединяющие большие узловы компьютеры. Передача данных между такими компьютерами осуществляется через спутники или специальные выделенные каналы связи. Все малые сети типа BBS могут являться абонентами этих глобальных сетей. Появляются и отечественные телекоммуникационные сети, наиболее крупными из них являются: ИАСНЕТ (IASNET), РЕЛКОМ, Исток-К, ИКС-МИР, ГЛАСНЕТ (GLASNET), Совам Телепорт (Sovam Teleport) и РИКО.

Пользователи и телекоммуникационные сети

После подключения к любой действующей компьютерной сети зарегистрированному в ней пользователю предоставляются разнообразные услуги, среди которых выделяются две главных:

- компьютерная межперсональная коммуникация (обмен сообщениями, электронные бюллетени новостей, телеконференции и т. д.);
- доступ к удаленным базам данных различного назначения.

Межперсональная коммуникация является самой распространенной и популярной среди пользователей телекоммуникационных компьютерных сетей, так

как предоставляет им возможность оперативной и качественной связи, а также передачи на большие расстояния, причем без потерь и искажений, данных в любом виде: текстов, таблиц, компьютерных программ, рисунков, фотографий и т.д. Помимо этого, по современным проводным телефонным сетям, имеющим большую пропускную способность, можно «крутить» телефильмы, передавать факс, звук и видеосигнал одновременно или производить через них миллионы операций в минуту.

И все же, поскольку общий принцип организации межперсональной компьютерной коммуникации схож с обычной почтовой связью, этот вид услуг принято называть электронной почтой. При этом каждый хост-компьютер сети превращается в большой почтамт, осуществляющий пересылку электронной почты и предоставляющий каждому своему абоненту его личный абонентский почтовый ящик, куда поступают все адресованные ему сообщения. Названия почтовых ящиков и сведения об их владельцах доступны всем абонентам сети.

Если несколько больших компьютерных сетей связаны друг с другом, их абоненты также могут обмениваться сообщениями.

Электронная почта завоевала в последнее время огромную популярность, поскольку она имеет большие преимущества в сравнении с традиционным обменом письмами по обычной почте, а именно:

- скорость доставки сообщений по электронной почте измеряется в минутах, а не в днях;
- сообщения или текст каких-либо документов можно одновременно отправлять большому количеству адресатов;
- входящую корреспонденцию можно хранить в центральном компьютере сети до тех пор, пока абонент не будет готов ее прочитать;

- отправка корреспонденции адресату осуществляется круглосуточно или в заранее заданное время;
- абонент автоматически получает подтверждение о том, что его сообщение было прочитано;
- полученные по электронной почте сообщения и информацию можно хранить на диске, редактировать при необходимости и распечатывать нужное количество копий на принтере.

В зависимости от количества участников переписки принято выделять три основные разновидности электронной почты:

- * простая электронная почта (один – одному);
- * почтовые списки (один – ко многим);
- * телеконференция (многие – ко многим).

Простая электронная почта обеспечивает отправку написанного (либо непосредственно в сети, либо подготовленного заранее) сообщения по нужному адресу и получение адресатом этого сообщения. При отправке письма можно дать системе команду подтвердить факт получения сообщения адресатом с указанием даты и времени получения. Немаловажно то, что система электронной почты гарантирует конфиденциальность передаваемой информации всем абонентам без исключения, поскольку почтовые телекоммуникационные системы являются закрытыми даже для системных операторов, управляющих ими, – это конструктивная особенность всех используемых пакетов программ.

Почтовые списки, использующиеся тогда, когда абонент хочет отправить сообщение или какой-либо документ одновременно нескольким адресатам, обеспечивают: организацию специальной службы переписки, ведение списка абонентов для рассылки корреспонденции, автома-

тическую рассылку сообщения по всем адресам списка. Почтовые списки эффективны для административно-управленческих структур, а также при регулярной рассылке коммерческой информации, реферативных статей или социологических опросников.

Самым сложным видом электронной почты является телеконференция. Телеконференция отличается от первых двух разновидностей электронной почты тем, что посвящена она обычно конкретной теме, вопросу, и при ее проведении очень большая роль отводится ведущему. Ведущий организует проведение телеконференции в сети, планирует и направляет ее работу, приглашает участников для обсуждения каких-либо проблем и вопросов и, подводя итоги работы, закрывает конференцию, когда необходимость в ней отпадает. Число участников конференции практически не ограничено, а время ее проведения — в среднем от 1–2 месяцев до полугода. Обычно в телекоммуникационной сети одновременно проводится несколько десятков конференций по различной тематике, и пользователь может участвовать в любой из них.

При проведении телеконференций автор какого-либо сообщения посылает его не конкретному адресату, а всем желающим его прочесть. Поэтому участие абонентов сети в телеконференциях аналогично созданию большого почтового ящика, к которому в удобное для них время могут обращаться несколько абонентов.

По такому же принципу организована разновидность телеконференций — электронная доска объявлений, или, как ее чаще называют в англоязычном варианте, BBS (Bulletin Board System). Она не случайно получила такое название, так как действительно похожа на обычную доску объявлений, где помещается любая информация, которая может представлять интерес для абонентов сети. В отличие же от телеконференций электронная доска объявлений действует постоянно, не так

строго структурирована по разделам, и ее создание не преследует какую-то конкретную цель.

В настоящее время компьютерные коммуникации позволяют каждому пользователю получить доступ к разнообразным базам данных, созданным на ЭВМ. За рубежом практически все публикации, включая коммерческую информацию, энциклопедии, реферативные и справочные издания, журналы и труды конференций, изначально готовятся в машиночитаемом виде и существуют одновременно как в традиционном виде в форме печатных изданий, так и на машинных носителях в форме баз данных. Базы данных зачастую объединяются в мощные банки данных.

Преимущество машиночитаемых баз данных заключается в том, что поиск информации в них осуществляется не за несколько часов или суток, а за считанные секунды. Причем, благодаря компьютерным коммуникациям с ними могут работать пользователи, находящиеся за тысячами километров от этих баз данных.

Без использования компьютерных коммуникаций для доступа к мощным базам данных трудно себе представить нормальное функционирование банков и бирж, коммерческих организаций и крупных предприятий, а также научного мира — НИИ, университетов и исследовательских центров.

Какие компьютерные программы используются при работе с телекоммуникациями?

Начиная работать в телекоммуникационной среде, каждый пользователь знакомится с двумя типами компьютерных программ: программами, которые используются для управления модемом в телекоммуникационной системе, имеющейся у пользователя, и программами, работающими на удаленном сетевом хост-компьютере, с которым пользователь поддерживает связь.

Для удобства работы пользователей с модемом были разработаны специальные коммуникационные программы, позволяющие автоматизировать стандартные процедуры соединения с удаленным компьютером, работы в сети и передачи данных, а именно:

- вести электронную телефонную книжку,
- осуществлять автоматический набор номера вызываемого удаленного компьютера,
- выбрать нужный режим передачи данных и протокол и многое другое.

Существует значительное число программ подобного типа, как отдельных программ (например, Procomm Plus, Bitcomm, MTE, QLink, Comit, Telix и др.), так и в составе интегрированных оболочек (например, в средах Framework, Microsoft Windows, PC Shell имеются небольшие коммуникационные программы, обеспечивающие пересылку обработанных данных на удаленный компьютер).

На сетевых хост-компьютерах установлены, как правило, специальные телекоммуникационные программы типа электронных досок объявлений, или BBS. Таких программ также создано огромное количество. Наиболее распространены сейчас такие BBS, как: Maximus-CBCS, PCBoard, Remote Access, TBBS, Wildcat!, PC Relay, Searchlight и т. д.

Функции большинства BBS сводятся к следующему:

- обеспечение одновременного доступа к главной машине сети нескольких пользователей,
- обеспечение обмена файлами и сообщениями с пользователями других сетей,
- предоставление пользователям доступа к информационным и справочным материалам и прочим ресурсам сети.

Какой модем выбрать?

Модемы выпускаются двух видов: как отдельные электронные платы, встроенные в компьютер (встроенные модемы), или как небольшие устройства в жестком корпусе, подключаемые к компьютеру через кабель (внешние модемы).

Основными производителями модемов для телефонных каналов являются компании США, Великобритании и Франции. Наиболее известные фирмы, выпускающие высококачественные модемы — Hayes Microcomputer Products, US Robotics, Multitech, Paradyne и ряд других. В последние годы очень активны на рынке модемов фирмы Южной Кореи и Тайваня.

Для того же, чтобы приобрести качественный модем, мало знать только название фирмы-производителя. Необходимо уточнить, как минимум, такие характеристики модема, как: скорость передачи данных, которую он поддерживает, Hayes-совместимость, а также наличие протокола коррекции ошибок.

Скорость передачи данных отражает количество двоичных сигналов (битов), передаваемых через модем в единицу времени, то есть измеряется скорость в битах в секунду. Наиболее распространены сейчас модемы со скоростью 1200, 2400 и 9600 бит/с. Максимальная скорость модемов на сегодня составляет около 38000 бит/с.

Очевидно, что чем выше скорость передачи данных, тем больший объем информации в единицу времени вы сможете передать по сети. С другой стороны, не все высокоскоростные модемы «выдерживают» устаревшее оборудование телефонных сетей в нашей стране. А кроме того, чем выше скорость передачи, тем больше вероятность появления ошибок в принимаемых данных, на что может повлиять даже небольшая помеха на линии связи или на АТС.

Для того чтобы при передаче данных избежать их искажения, используют специальные наборы семантических и син-

тактических правил, определяющих поведение функциональных блоков данных, которые называют сетевыми протоколами. При покупке модема следует обращать внимание на то, поддерживает ли модем стандартный протокол коррекции ошибок MNP (Microcom Networking Protocol).

Протокол MNP непрерывно совершенствуется, и в настоящее время существует 10 классов (уровней) этого протокола. Начиная с класса 5, протокол не только позволяет корректировать ошибки при передаче данных, но и реализует функцию сжатия данных, что заметно убыстряет процесс передачи и делает его более эффективным.

Протоколы коррекции ошибок бывают «встроены» в модем, т.е. находятся непосредственно в памяти модема и запускаются без каких-либо специальных действий со стороны пользователя, или эмулируются дополнительно с помощью специальных телекоммуникационных программ (например, MTE, Comit и др.). Но в любом виде они выполняют свою главную функцию — обеспечивают отсутствие искажений при передаче информации.

Особенно много проблем у пользователей возникает с Hayes-совместимостью модема, то есть способностью модема выполнить определенный стандартный набор команд, разработанный фирмой Hayes Microcomputer Products. Для обеспечения полной совместимости модем должен не только поддерживать полный набор команд Hayes, но и отвечать компьютеру сообщениями, которые с точностью до буквы соответствуют сообщениям, выдаваемым эталонным модемом Hayes (Smartmodem).

При «общении» компьютера и модема используется особый язык, представляющий собой набор команд. Большинство современных модемов используют набор команд Hayes, названных по имени одного из первых производителей модемов Дени-

за Хейза (Dennis Hayes) — представителя фирмы Hayes Microcomputer Products.

Большинство Hayes-команд предваряется двумя символами — буквами AT. Эта последовательность называется символом внимания (attention character), например: ATDP1234567, где AT — символ внимания, D — (Dial — «звонить»), P — (Pulse — «пульсовым методом»), 1234567 — телефонный номер удаленного компьютера.

Часто набор Hayes-команд называют также AT-командами. Модем, распознающий AT-команды, называется Hayes-совместимый модем.

Hayes-совместимость модема, т.е. способность модема выполнять стандартный набор команд, ставший уже международным, это очень важная характеристика модема, так как только Hayes-совместимые модемы позволяют свободно работать и в крупных отечественных, и в зарубежных телекоммуникационных сетях в отличие от таких широко разрекламированных отечественных модемов, как «Лександ», «Мастак» или «Unicom», «говорящих» на своем языке команд и поэтому позволяющих связываться только с подобными модемами.

На всех современных Hayes-совместимых модемах (или в документации к ним) обязательно указываются международные протоколы связи (принятые в зарубежных и отечественных сетях), которые данные модемы поддерживают. Например, для модемов, работающих на скорости 2400 бит/с, существует протокол v.22, v.22 bis, а для модемов на 9600 бит/с — v.32 и т.д.

Модемы работают обычно либо в дуплексном, либо в полудуплексном режиме передачи данных.

При дуплексном режиме данные передаются через модем одновременно в обоих направлениях. Это значит, что нажатие клавиш немедленно отобразится на экране, тем самым исключаются бесконечные задержки из-за переключения направления передачи данных.

При полудуплексном режиме данные передаются в каждый момент времени только в одном направлении. Этот режим можно сравнить с игрой в пинг-понг: данные передаются в разные стороны по очереди. Эта схема удобна в тех случаях, когда нужна только односторонняя передача данных (телефаксы, передача файлов), но не подходит для интерактивного доступа (как, например, в BBS).

Большинство современных модемов работает в дуплексном режиме.

Выбор телекоммуникационной сети

Если вы хотите работать в какой-либо телекоммуникационной сети, но не знаете, какой сети отдать предпочтение, соберите сначала информацию о всех телекоммуникационных сетях, имеющихся в вашем городе, регионе, республике, обращая внимание на тематическую направленность (особенно раздела телеконференций) сети и то, кто является ее участниками. Также немаловажный вопрос, обеспечивает ли данная сеть пересылку корреспонденции в зарубежные сети. Так, например, очень многие российские школы самостоятельно «вышли» и зарегистрировались в таких телекоммуникационных сетях, как РЕЛКОМ и ГЛАСNET, являющихся в настоящий момент крупнейшими неспециализированными сетями электронной почты с разветвленной системой телеконференций, действующими на территории нашей страны.

Большинство компьютерных телекоммуникационных сетей в мире, в том числе и в России, являются коммерческими, и пользование услугами таких сетей является «дорогим удовольствием» для обычных государственных школ, не имеющих дополнительных источников финансирования, да и по тематической направленности эти сети не всегда представляют интерес для учителей и учащихся.

В последние годы за рубежом были созданы специальные некоммерческие компьютерные телекоммуникационные сети, объединяющие, как правило, университеты, школы и организации, напрямую с ними связанные (библиотеки, научные лаборатории, научные общества и т. д.), например, сеть NGS Kids Network и World Classroom в США, CAMPUS 2000 в Великобритании, PEGASUS в Австралии.

Если выбор сделан, узнайте, где находится штаб-квартира интересующей вас сети или ее региональное представительство. Далее изучите тарифы на услуги сети и выясните, нет ли для вас каких-либо льгот (как, например, для бюджетной организации, школы и т. п.). После этого заключается договор с представителями администрации сети на абонентское обслуживание (на месяц, квартал или год), в результате чего каждому абоненту присваивается определенное сетевое имя (набор символов, под которым он будет зарегистрирован в сети и внесен в базу данных пользователей) и проводится процедура регистрации.

Время работы в сети обычно оговаривается в процессе заключения договора. Чаще всего используется принцип «счетчика» — повременной тариф, поэтому стоимость работы в сети напрямую зависит от времени работы в режиме on-line и активности почтовой переписки.

До начала работы в сети стоит изучить руководство пользователя, из которого вы получите исчерпывающую информацию об основных режимах работы сети и ее особенностях.

Если в процессе работы в сети у вас возникнут какие-либо проблемы, вы можете обратиться к системному оператору сети (Sysop, System Operator), контролирующему обычно работу пользователей в сети и поддерживающему все функции, выполняемые главным компьютером сети.

М.Л. Гуткин

ИНФОРМАТИЗАЦИЯ ОБРАЗОВАНИЯ И КОМПЬЮТЕРЫ MACINTOSH

Внедрение компьютеров в сферу образования связывалось с информатизацией образования, с преподаванием компьютерной грамотности, с введением курса «Основы информатики и вычислительной техники». Исходя из этого и решались проблемы, связанные с разработкой и производством учебных ПЭВМ и КУВТ на их основе. Считалось, и частично для того времени было справедливым, что в качестве учебных ПЭВМ можно использовать упрощенные модели профессиональных ПЭВМ (т.е. ПЭВМ с минимальными характеристиками по быстродействию, объемам оперативной и внешней памяти и т.п.). При этом основной упор делался на обучение работе с компьютером: алгоритмизация, программирование и освоение работы с пакетами прикладных программ (обработка текстов, базы данных, электронные таблицы), что, в свою очередь, служило ориентиром для разработки учебных ПЭВМ и КУВТ отечественного производства и для централизованных закупок импортных ПЭВМ [1]. Для образовательных учреждений в рамках «Пилотного проекта» была осуществлена закупка моделей минимальной конфигурации персональных ЭВМ серии PS/2 фирмы IBM.

В настоящее время ситуация резко изменилась и характеризуется, на наш взгляд, следующими основными моментами.

Первоначальный этап компьютеризации образования уже пройден. Разработано программное обеспечение, подготовлены преподаватели и учителя соответствующей квалификации, написаны учебники, во многих школах имеется вычислительная техника.

Индустрия производства отечественных ПЭВМ переживает тяжелый период. «Выжившие» предприятия перешли на отверточную сборку IBM PC-совместимых компьютеров, что во многом определяет тип ПЭВМ, поставляемых сейчас в систему образования России.

Наблюдается переход от централизованной финансовой политики закупки и распределения учебных ПЭВМ к решению этой проблемы на региональном уровне. Как следствие, у учебных заведений появляется возможность выбора.

Пришло понимание, что ПЭВМ является мощным инструментом, существенно повышающим эффективность учебного процесса. В связи с этим изменились и требования к характеристикам учебных ПЭВМ. Теперь это не упрощенный компьютер, а профессиональная ПЭВМ, ориентированная на вполне конкретную сферу применения – образование.

В образовании начинают активно внедряться системы мультимедиа и сетевые технологии.

К системам мультимедиа относятся специальные программно-аппаратные средства, расширяющие возможности ПЭВМ и рабочих станций, которые позволяют обрабатывать не только алфавитно-цифровую информацию, но и звук, и видео (неподвижные изображения, «живое» видео, воспроизводящее на дисплее ПЭВМ видеофильмы, а также двух- и трехмерная анимация) [2].

Сетевые технологии включают использование локальных сетей и средств телекоммуникации [3].

При внедрении систем мультимедиа в

образование на первый план выступают следующие проблемы:

- *методическая проработка новых педагогических технологий, определение наиболее оптимального с точки зрения учебного процесса сочетания различных видов информации (текст, графика, видео, звук);*
- *наличие таких авторских средств, которые позволяют учителю (преподавателю)–непрофессионалу в области вычислительной техники самому создавать и модифицировать как отдельные уроки (лекции), так и целые учебные курсы. Для этого необходим переход от традиционных операционных систем (MS DOS) и от интерфейсных надстроек (WINDOWS) к мощным авторским средствам, обеспечивающим простоту манипуляции со всеми видами информации. Учитель должен оставаться в привычной для себя среде и в то же время иметь возможность взаимодействия с системами мультимедиа;*
- *наличие ПЭВМ, на которых, с одной стороны, реализованы средства мультимедиа, а, с другой стороны, упомянутые выше авторские средства;*
- *наличие методических рекомендаций по использованию как этих компьютеров, так и авторских средств;*
- *реализация сетевых технологий (по крайней мере, на уровне локальных сетей), позволяющих легко объединять компьютеры в сеть и использовать на этих компьютерах системы мультимедиа.*

Наиболее оптимально перечисленным выше требованиям отвечают ПЭВМ-линии Macintosh фирмы Apple. Ведь именно с компьютеров фирмы Apple и началась эпоха персональных ЭВМ. Рассмотрим коротко их основные особенности.

В первую очередь следует отметить,

что Macintosh и его базовое программное обеспечение разрабатывались «одной командой», что весьма благоприятно отразилось на результатах. Создатели Macintosh в первую очередь заботились о пользователе, о том, чтобы с компьютером было легко работать. К будущему компьютеру были выдвинуты два основных требования:

- *человек, работающий с Macintosh, должен оставаться в привычной для себя среде (папка с бумагами, карта, записная книжка и т.п.);*
- *для управления компьютером должно быть достаточно одной кнопки.*

Кроме того, особое внимание было уделено графическим средствам, а также обеспечению возможности простой переконфигурации компьютера при подключении новых периферийных устройств.

В результате был создан компьютер с исключительно удобным интерфейсом, высокой производительностью и с графическими средствами, существенно опередившими другие модели ПЭВМ.

Перечислим основные особенности ПЭВМ линии Macintosh:

- *для установки и подключения компьютера требуется минимум знаний;*
- *подключение компьютера к локальной сети также исключительно просто – достаточно лишь воткнуть соединительный кабель (все программные средства присутствуют изначально, никакой генерации сети не требуется);*
- *в корпус компьютера заложена возможность крепления к столу, что практически необходимо при эксплуатации в учебном классе;*
- *внутренний блок питания обеспечивает работоспособность компьютера при любом значении напряжения питания в диапазоне от 100 до 240 В;*
- *для защиты от случайных выбросов в*

сети питания блок питания имеет электронный предохранитель, который отключает компьютер, если напряжение выходит за допустимые пределы. При восстановлении нормального напряжения компьютер снова становится работоспособным;

- в мониторах используются только высококачественные дисплеи Тринитрон фирмы Сони, что обеспечивает высокие потребительские качества мониторов (разрешение не менее 640x480 при 256 цветах, частота 70 Гц – это соответствует нашим санитарно-гигиеническим нормативам);
- продуманность интерфейса работы с «мышью» снижает нагрузку на клавиатуру, которая практически нужна только для ввода текстовой информации – все управление осуществляется посредством «мыши»;
- компьютеры легко расширяются путем подключения дополнительных устройств через интерфейс SCSI, причем для большинства стандартных устройств не нужно регенерировать систему – достаточно только выполнить перезагрузку (система сама при включении или перезагрузке определяет свою конфигурацию);
- использование графических шрифтов позволяет легко адаптировать компьютер к любым национальным алфавитам;
- все компьютеры семейства имеют встроенный вход/выход для звука, а также встроенный высококачественный динамик, что вместе с графикой без дополнительных затрат позволяет строить и использовать средства мультимедиа;
- пользовательский интерфейс построен

таким образом, что нет необходимости изучать операционную систему, ее команды и т.п. Благодаря системе окон пользователь сразу может работать с компьютером, обладая минимумом знаний о нем. Реально – это работа в привычной для пользователя среде.

Компьютеры Macintosh выпускаются только на заводах фирмы Apple, поэтому снимается проблема цветной сборки. Кроме того, фирма предъявляет высокие требования к своим дилерам и к сервисной службе; компьютеры поставляются только в те регионы, где возможно их качественное и быстрое обслуживание. Этим обеспечивается высокое качество эксплуатационного обслуживания (в котором компьютеры нуждаются в минимальной степени). Более подробно об использовании этих компьютеров в образовании мы расскажем в следующей публикации.

Литература

- [1]. Гуткин М.Л., Иванов А.В., Новосельцев С.К., Христочевский С.А. Персональные ЭВМ учебного назначения// Информатика и образование. – 1990. – №6.
- [2]. Новосельцев С.К. Мультимедиа – синтез трех стихий// Компьютер Пресс. – 1991. – №7, 8, 11.
- [3]. Блэк Ю. Сети ЭВМ. – М.: Мир, 1990.
- [4]. Новосельцев С.К. Мультимедиа-91: достижения, тенденции, рынок// Компьютер Пресс. – 1992. – №7.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИИ
ЦЕНТР ИНТЕРАКТИВНЫХ СРЕДСТВ ОБУЧЕНИЯ

ПРЕДЛАГАЕТ:



Широкий выбор
программно-методических комплексов по различным предметам школьного курса для IBM-совместимой

техники, снабженных методическими и дидактическими материалами:

- информатика
- математика
- физика
- астрономия
- химия
- история
- география, биология и экология
- русский язык
- английский язык

Инструментально-педагогические средства для изготовления контрольно-обучающих программ, систем моделирования, составления расписания, тарификации учителей, контроля успеваемости.



Учебные классы на основе IBM-совместимых персональных компьютеров.

1. PS/1 мод 241/286/1M6/40M6/FDD 3.5"/VGA класс = 11 ПЭВМ $\Sigma = \$10745$
2. Сервер AT386/SX-33/2M6/120M6/"косая"/SVGA рабочая станция AT386/SX-33/2M6/SVGA класс = сервер + 10 п.с. + сеть Ethernet $\Sigma = \$9746$
3. Сервер AT386/SX-33/2M6/120M6/"косая"/SVGA рабочая станция AT286/SX-20/1M6/SVGA класс = сервер + 10 п.с. + сеть Ethernet $\Sigma = \$8096$
4. Принтеры Epson, Citizen, HP
5. CD-ROM проигрыватель + контроллер $\Sigma = \$555$



Новый Курс Информатики

создан, апробирован, внедрен в более чем 1000 школах бывшего СССР в ходе экспериментального образовательного проекта "Пилотные школы", проводимого совместно с фирмой IBM.

Предлагаемый программно-методический комплекс делает этот курс доступным не только для школ, имеющих IBM-совместимую технику, но и огромному числу школ, оснащенных классами УКНЦ.

Комплекс включает:

- *Программы* Schoolworks, Roo & Robby и системную часть ("быстрая сеть", аналог Norton Commander), а также решения и комментарии к более чем двумстам задачам курса;
- *Двенадцать задачников* с условиями задач курса;

- *Двенадцать учебников* "Основы информатики и вычислительной техники" Гейн А. Г. и др.;

- *Книгу учителя*, включающую методические материалы и тематические поурочные планы.

Цена комплекса: \$19

Оплата в рублях по текущему курсу ММВБ.

Ждем Вас по адресу: 125315, Москва, ул. Часовая, 216.

Расчетный счет 1609325 в Ленинградском отд. Мосбизнесбанка МФО 201694.

тел.: (095) 155 8730, (095) 155 8737,

факс: (095) 155 8727.

ПЕДАГОГИЧЕСКИЙ ОПЫТ

М. Д. Карпов

УНИВЕРСАЛЬНАЯ ТЕСТ-ПРОГРАММА ДЛЯ ПРОВЕРКИ ЗНАНИЙ В ШКОЛЕ

Идея создания этой программы возникла после рассмотрения некоторых тест-программ. Естественно, она не заменит сложную, хорошо использующую разные средства (в том числе графику), обучающую программу, но организовать контроль, а также коррекцию знаний и даже обучение она поможет. При этом нет необходимости каждый раз заново писать программу, достаточно менять блок данных или тест-модуль.

Для этого из набора тестов нужно сделать этот модуль и скомпоновать его с рабочим (основным) модулем по описанным ниже правилам. Затем проверить работу полученной таким образом тест-программы (и если нужно – исправить ошибки). То, что получилось, записывается на дискету под соответствующим именем, и в вашем распоряжении готова тест-программа по конкретной теме!

Но несмотря на простоту, в программе заложена возможность менять не только тему (например, изучаемый предмет: литература, русский или иностранный язык, география, химия и т. д.), но и вид теста. Пассивный – когда даются варианты ответов, или активный – когда ответ должен сформулировать сам тестируемый.

Возможна комбинация активных и пассивных тестов в одном тест-модуле.

Ниже приводится описание самой программы, точнее ее рабочего (основного) модуля (приложение 1). Описывается процедура компоновки, а также дается пример тест-модуля (приложение 2).

UNTEST – программа, предназначенная для проверки знаний учеников при помощи тестов.

Она универсальна. Построена по модульному принципу и состоит из двух частей: основной, или рабочий, модуль и тест-модуль. В тест-модуль записывают: количество тестов, время ответа, «критерий строгости», название темы и правила работы. Далее – данные теста, варианты правильных ответов и некоторая другая информация.

Работа с программой строится по следующему плану. Отдельно составляется тест-модуль и присоединяется к рабочему модулю.

Это делается так: с РМП вводится основной модуль на БК-0010, а к нему в режиме редактирования добавляется тест-модуль, т. е. тесты и другие данные.

Они пишутся, начиная с 2000-й строки в следующем формате:

2000 DATA <количество вариантов>%,
<время на ответ(с)>%,<критерий строгости>%
Пример 1: 2000 DATA 10%,30%,6%

Это означает, что вы задаете 10 вариантов тестов, 30 с времени на ответ, 6 – означает, что при 6 ошибках оценка будет «2».

В строках 2010–2999 (номера строк произвольны в этом интервале) вводятся: предмет, по которому ведется тест; тема теста и правила работы с ним для учеников. Для них отведено четыре места (четыре символьные переменные). Два для темы и два для правил. Запись темы показана в примере 2.

Пример 2: 2010 DATA " Английский язык "
2020 DATA " Глаголы to do, to be "

Потом в строках, например 2030 и 2040, вы таким же образом записываете правила работы с тестом для учеников. При этом не забудьте, что максимально вы можете записать в одной строке 255 знаков, включая номер строки и слово DATA.

Если же вам правила не нужны, то вы должны записать «пустой» текст:

2030 DATA "", ""

Варианты тестов записываются с 3000-й строки в следующем формате:

3000 DATA "<текст варианта>",
"<правильный ответ (номер или текст)>",
<место ответа в варианте>

Таким способом записываются все варианты, количество которых вы указали в строке 2000. Естественно, только первый имеет номер строки 3000, остальные записываются под любыми большими.

Если правильный ответ необходимо вставить в текст варианта после отработки, то в этом тексте место для вставки нужно обозначить пробелами и указать его числом, считая от начала текста каждый

знак, включая пробел, а также невидимые на экране знаки перевода на новую строку, если вы их использовали (см. примеры 3, 4, 5).

Пример 3: 3000 DATA "Я был уверен поезд
мчит меня к счастью.",":",13%

Это означает, что в заданном варианте на 13-м месте (между словами «уверен» и «поезд») будет поставлен знак «:».

Пример 4: 3040 DATA " What he see ?",
"does",6%

В этом примере между словами «What» и «he» будет вставлено слово «does». Начало его с 6-го знака. Еще пример:

Пример 5: 3030 DATA "КАРОВА","O",2%

Здесь дается задание исправить ошибку. Второй буквой должна стоять буква «O».

Это примеры активных тестов. То есть правильный ответ в данных есть, но на экране во время тестирования его не будет. Тестируемый сам должен его вспомнить и набрать на клавиатуре.

Если ответ не нужно вставлять в текст варианта, то элемент место ответа в варианте указывается равным нулю. Смотрите пример 6.

Пример 6: 3040 DATA "sehen читать
плавать
смотреть
сидеть
писать",":3",0%

В этом примере вставка не нужна. Нужно только указать правильный вариант ответа – третий.

Это тест пассивный, так как на экране будут и варианты ответов.

После подготовки данных теста (набора всех строк в требуемом формате) нуж-

но проверить работу программы и записать ее на дискету.

Обратите внимание! Чтобы не стереть основной рабочий модуль (он понадобится для создания других тестов), записывайте сделанную программу под другим именем или на другую дискету. Лучше и то, и другое.

После этого вы имеете готовую тест-программу по конкретной теме.

Проводите тестирование!

При работе учеников с тестом они набирают ответ на клавиатуре и вводят его клавишей ВВОД. Набираемый ответ отображается на экране.

Имеется возможность исправления вводимого ответа. Исправление только клавишей ЗБ! На БК она находится под клавишей СТОП.

Следует иметь в виду, что исправление возможно до нажатия клавиши ВВОД. Время набора и исправления включается во время, отведенное для ответа в строке 2000. Если вы не успели нажать клавишу, на экране возникает надпись «Ваше время истекло» и вариант правильного ответа. Правильный ответ выводится и в случае, если ответ ученика неверен. Ведь цель программы – не только заметить ошибку, но и закрепить правильный ответ.

Тестируемому не нужно вставлять ответ в соответствующее место текста на экране. Ему нужно просто набрать его на клавиатуре и нажать клавишу ВВОД. Вставка осуществляется автоматически.

Дополнение 1. *Величина «критерия строгости» может быть от «4» до «7». Она показывает, что если ученик совершает такое или большее количество ошибок, то он получает «2» (если он не успевает ответить за отведенное время, то это тоже рассматривается как ошибка). «5» ученик*

получает только в случае отсутствия ошибок при любом значении «критерия».

Дополнение 2. *Для КУВТ-86 есть возможность простой защиты от «подсматривания». Для этого нужно вставить в уже готовую для работы программу (ни в коем случае не в исходный рабочий модуль) строку:*

7 РОКЕ &O202,&O36000

В этом случае после ввода и запуска (!) программы ее листинг просмотреть невозможно. Немедленный запуск программы после ввода обеспечивает тот, кто проводит тестирование.

Если ученики тестируются впервые и эта процедура им непривычна, то, конечно, нужен предварительный инструктаж.

Дополнение 3. *Данная программа с небольшой доработкой может быть реализована и на другой технике (в других версиях Бейсика). При этом в первую очередь необходимо обращать внимание на форму операторов ввода-вывода, обозначение переменных символьных и целых типов. Кроме того, в этом случае количество времени, даваемого на ответ, может не соответствовать количеству секунд, указанному в тест-модуле, так как этот параметр зависит от быстродействия.*

Естественно, может измениться сама процедура включения в программу тест-модуля. Описанная выше отражает особенности КУВТ-86.

По этой же причине ошибочным будет включение в программу строки защиты от «подсматривания» (см. дополнение 2).

Дополнение 4. *Программа предусматривает вывод 32 символов в строке дисплея. Если вывод иной – необходимо*

скорректировать строки 1220 и 1240 программы.

Иллюстрирует применение описанных правил и работу программы модуль тестов для проверки знаний по расстановке зна-

ков препинания в бессоюзном сложном предложении (приложение 2). Он уже апробирован. Автор тестов – преподаватель русского языка Снегирева И. В.

ПРИЛОЖЕНИЕ 1. Рабочий модуль тест-программы UNTEST.

```

10 CLEAR 2000
12 RESTORE 2000
21 READ PP%
22 READ NT%
23 READ ND%
24 READ Z1$
25 READ Z2$
26 READ Z3$
27 READ Z4$
28 RESTORE 3000
80 CLS
90 PRINT AT(4,7);Z1$
100 PRINT AT(2,12);Z2$
110 GOSUB 1100
115 GOSUB 1100
120 CLS
130 PRINT AT(2,1);Z3$
135 PRINT AT(2,5);Z4$
140 PRINT AT(3,18); "Нажмите любую клавишу!"
150 KL$=INKEY$
160 IF KL$="" GOTO 150
170 NW%=0
180 NP%=0
183 PX%=-3
186 PY%=-8
190 CLS
195 PRINT AT(3,4); "Первый вариант"
200 FOR I%=-1% TO PP%
203 READ P$
205 READ AN$
208 READ PL%
220 PRINT AT(PX%,PY%);P$
230 PRINT AT(1,16);"Ваш вариант ответа ?"
233 PRINT AT(10,19);"Ответ ?"
235 GOSUB 1200
240 NZ%=0
245 VAR$=""
250 KL$=INKEY$
260 NZ%=NZ%+1%
270 IF NZ%=-38%*NT% GOTO 500
275 IF KL$="" GOTO 250
279 IF ASC(KL$)24% GOTO 282
280 VAR$=MID$(VAR$,1%,LEN(VAR$)-1%)
281 GOTO 284
282 IF ASC(KL$)-10% GOTO 295
283 VAR$=VAR$+KL$
284 PRINT AT(17,19);VAR$;" "
287 GOTO 250
295 IF VAR$AN$ GOTO 400
300 PRINT AT(1,16); "Ответ верный. МОЛОДЕЦ!"
310 IF PL%=-0% GOTO 320
315 PRINT AT(ZX%,ZY%); AN$
320 GOSUB 1100
330 GOSUB 1100
335 GOSUB 1100
340 GOTO 440
400 NW%=NW%+1
410 PRINT AT(1,16); "Ответ неверный!
      Ошибка вышла."
420 PRINT AT(3,19); "Правильный ответ ";AN$
430 GOTO 310
440 CLS
450 PRINT AT(3,4);"Следующий вариант"
460 GOTO 600
500 NP%=NP%+1
510 PRINT AT(1,16);"Ваше время истекло! "
530 GOTO 420
600 NEXT I%
610 GOSUB 1000
620 CLS
630 PRINT AT(3,3);
      "Всего попыток: "; PP%
640 PRINT AT(3,4);
      "Правильных ответов: "; PP%-NP%-NW%
650 PRINT AT(3,5);
      "Неправильных: "; NW%
660 PRINT AT(3,6);
      "Отсутствие ответов: "; NP%
670 PRINT AT(5,10);
      "ВАША ОЦЕНКА: "; MARK%

```



```

700  END
1000 NR%-NW%+NP%
1010 IF NR%=0% THEN MARK%=-5%
1020 IF ND%=-4% AND NR%=-1% THEN
MARK%=-4% ELSE IF ND%4% AND
(NR%=-1% OR NR%=-2%) THEN MARK%=-4%
1030 IF ND%=-4% AND (NR%1%AND NR%3%)
THEN MARK%=-3% ELSE IF
ND%4% AND (NR%2% AND NR%) THEN
MARK%=-3%
1040 IF NR%=ND% THEN MARK%=-2%
1050 RETURN
1100 FORT=0 TO 300
1110 X=17/19
1120 NEXT T
1130 RETURN
1200 ZX%=PL%+PX%-1
1210 ZY%=-PY%
1220 IF ZX% GOTO 1260
1230 ZY%=-ZY%+1
1240 ZX%=-ZX%-32
1250 GOTO 1220
1260 RETURN
    
```

ПРИЛОЖЕНИЕ 2. Модуль тестов по теме «Знаки препинания в бессоюзном сложном предложении».

<p>2000 DATA 10%.30%.6%</p> <p>2010 DATA "Зачетная практическая работа по русскому языку по теме", "ЗНАКИ ПРЕПИНАНИЯ В БЕССОЮЗНОМ СЛОЖНОМ ПРЕДЛОЖЕНИИ"</p> <p>2020 DATA "Определите, что нужно поставить между частями БСП: запятую, точку с запятой, двоеточие или тире."</p> <p>2030 DATA "В случае неправильного ответа повторно прочитайте предложение с правильно расставленными знаками препинания"</p> <p>3000 DATA "А просьба моя состоит в следующем берегите наш язык.",",",34%</p> <p>3005 DATA "Встречались мы с дождем и выюгами огонь и воду вместе мы прошли.",",",34%</p> <p>3010 DATA "Андрей улыбнулся и осторожно положил трубку на рычаг послышались короткие гудки.",",",53%</p> <p>3015 DATA "Я преклоняюсь перед правилами люблю порядок.",",",30%</p> <p>3020 DATA "Я чувствую уловил желанный тон для своей лесной повести.",",",11%</p> <p>3025 DATA "Пахнет в хате квашней и дымом за порогом бурлит весна.",",",30%</p> <p>3030 DATA "Спать ложился дядя Степа ноги клал на табурет.",",",25%</p> <p>3040 DATA "Ледоход уже начинался где-то</p>	<p>дальше, выше по реке,лед напирал,давил, подгонял еще не тронувшиеся поля.",",",22%</p> <p>3050 DATA "Догорал апрельский светлый вечер по лугам холодный сумрак лег.",",",33%</p> <p>3060 DATA "Я был уверен поезд мчит меня к счастью.",",",13%</p>
--	--

Примечания редактора

1. В программе не предусмотрен случайный выбор вопросов из запрограммированного списка. Вместо этого осуществляется простой перебор всех заданий от первого до последнего при каждом новом запуске.

2. После вывода на экран сообщения о назначении теста (текст в операторе DATA в строке 2010), а также после ответа на очередной вопрос и вывода сообщения «Ответ верный» или «Ответ неверный» следует ждать вывода следующего вопроса, не нажимая никаких клавиш во избежание сбоев.



**БЮРО
ИНТЕЛЛЕКТУАЛЬНЫХ
ТЕХНОЛОГИЙ**

Модем 1200 бод

с программным обеспечением

- программная реализация протоколов V.23 и V.21
- скорость связи 1200, 600, 300 бод
- совместим с
ССТП V.23 1200 бод
ССТП V.23 600 бод
ССТП V.21 300 бод
BELL 103 300 бод
- удобная в обращении программа-терминал для связи УКНЦ-УКНЦ, УКНЦ-IBM
- межшкольная BBS (электронная доска объявлений)
- подробное описание

Наш модем это уникальная возможность использовать УКНЦ для выхода в глобальные компьютерные сети

УКНЦ

Auto Control

конструктор тестов

- удобный интерфейс
- разнообразие типов вопросов
- выбор критерия оценки ответов
- установка способа выбора вопросов из курса
- сбор и распечатка статистики
- возможность создания графических кадров
- генерация исполнимой тестовой программы в формате EXE
- система контекстных подсказок
- подробно составленное руководство

Эти возможности делают Auto Control наилучшим выбором среди контролирующих систем

IBM

Constructive Geometry

учебная среда

Пакет включает набор открытых для пользователя модулей:

- построения циркулем и линейкой
- сечение многогранников
- геометрические преобразования на плоскости
- элементы теории групп
- наглядная кристаллография
- геометрический debugger
- база данных геометрических задач

Каждый модуль может быть "наполнен" задачами, комментариями, подсказками и решениями, содержащимися в поставке, либо подготовленными преподавателем

IBM

Control CAD

конструктор тестов

- встроенный графический редактор
- удобная интегрированная среда
- многочисленные параметры теста и типы вопросов
- конечным продуктом является выполняемая программа (.SAV)
- результаты сохраняются в файл на машине преподавателя
- специальная программа обработки и печати результатов тестирования

С помощью ControlCAD Вы сможете подготовить любые тесты

УКНЦ

Blank Editor 1.0

новый редактор бланков для школьной канцелярии

- многооконный интерфейс
- система контекстной помощи
- встроенный текстовый редактор с русификатором
- поддержка мыши
- позволяет создавать и редактировать бланки любой сложности
- запись заполненных бланков в файл или вывод на принтер
- хранение ранее введенной информации в базе данных
- набор готовых бланков для школьной канцелярии и бухгалтерии

Наша программа возьмет на себя рутинную работу и сэкономит Ваше время

IBM

Пакеты для поддержки курсов начальной школы

Мозаика
Матрис1 Матрис2 Матрис 3
Сетевое Лото
Роботландия с новой сетью SERVER
Натуральные числа

Учебные средства

пакет "Математика"
Графики функций
физический практикум Механика
обучающий пакет по информатике Сократ
язык программирования Mif Logo
Учебная база данных и Учебные электронные таблицы
пакет "Машинопись" и клавиатурный тренажер Touch Typing
пакеты "Си для начинающих" и "Паскаль для начинающих"
графическая система для подготовки тестов ControlCad

Прикладные программы

командная оболочка PAF Commander
программа сервисной печати документов UK Print
утилита T-Formatter
универсальный форматтер и копировщик дискет NEWFMZ

Инструментальные программы

пакет UKGraph для работы с периферийным процессором
усовершенствованный компилятор языка Си
компилятор языка Форт
среда разработки Fast Pascal

Деловые применения

система подготовки документации Мини-Канцелярия
система подготовки документации Мини-Бухгалтерия
автоматизированная система Склад & Реализация
графический редактор КвантUG и сервис к нему UGPrint
графический редактор Fast Draw
текстовые редакторы Writer и Ассистент

Игровые программы

пакеты оригинальных игр (от Игрушки-1 до Игрушки-6)
программный эмулятор БК-0010 (монитор локальной сети + игры)

Аппаратные средства

электронные диски 512kb с монитором локальной сети
программно-аппаратный комплекс "Секретарь"
модем УК-1200
манипулятор типа "мышь"

**Неш !!! Плоттер для УКНЦ
с программным обеспечением**

**ПРОГРАММНЫЕ И АППАРАТНЫЕ КОМПЛЕКСЫ
ДЛЯ УКНЦ**

НАШ АДРЕС:

103498 Москва, К-498, а/я 164

ТЕЛЕФОНЫ:

535-22-22 534-48-31 534-48-32 (факс)



ПОСТАВКА УЧЕБНЫХ КОМПЬЮТЕРНЫХ КЛАССОВ НОВОГО ПОКОЛЕНИЯ

Распространенная конфигурация компьютерного класса на базе IBM AT 386 и 486:

- ✓ сервер локальной сети (AT 386 или AT 486)
- ✓ рабочий компьютер учителя (AT 386 или AT 486)
- ✓ 10 ученических машин (AT 386 или AT 486)
- ✓ оборудование для локальной сети
- ✓ сетевое программное обеспечение (Novell NetWare или lola)
- ✓ базовый набор системного программного обеспечения
- ✓ набор обучающих программ по школьному курсу
- ✓ полный комплект программ по курсу информатики
- ✓ принтер
- ✓ модем
- ✓ сканер
- ✓ звуковой адаптер

Конфигурация компьютерного класса может быть подобрана по Вашему желанию
Цена класса от \$8000, оплата в рублях по курсу

ГАРАНТИЯ В ТЕЧЕНИЕ 1 ГОДА

АРМ "Администратор":

- ✓ компьютер IBM AT386 или IBM AT486
- ✓ принтер
- ✓ Бухгалтерия школы + Зарплата для школы
- ✓ База данных "Школьная документация"



Обучение и информационно-техническое обслуживание
поставки комплектующих для компьютеров типа IBM
дооснащение и модернизация компьютерных классов

Доставка в любую точку России

НАШИ ДИЛЕРЫ, РАСПРОСТРАНЯЮЩИЕ ПРОГРАММНЫЕ ПРОДУКТЫ

- Н.Новгород, (8312)-68-75-14, (8312)-68-48-39
- Орел, МП "Колор", 086-00-5-18-03, 086-00-2-87-87
- Свердловская обл., г. Качканар, Межшкольный компьютерный центр, (34341)-2-24-63
- Киев, Фонд программных средств обучения, 244-72-04, 244-94-85
- Днепропетровск, Евротехком, 93-39-14
- Тольятти, Педагогический институт, каф. геометрии, 35-27-58
- Владивосток, Центр "Информатика в образовании", 22-94-79

534-48-32

534-48-31

535-22-22

А.Г. Юдина

БЕЙСИК ИЛИ ПАСКАЛЬ? ЛОГО, КОНЕЧНО, ЛУЧШЕ!

Лого — язык программирования и вместе с тем особая обучающая среда. Разработали Лого ведущие американские исследователи в области искусственного интеллекта. Язык этот по синтаксису предельно прост и близок к естественному. В то же время он обладает мощными современными средствами, формирующими культуру мышления и позволяющими создавать программы очень лаконичные, прозрачные по структуре и эффективные.

Лого — замечательное средство для моделирования чего угодно. В вашем распоряжении будут от одного до четырех исполнителей-черепашек, которые могут менять свою форму, создавать рисунки, двигаться по любым траекториям с разными скоростями, сообщать вам данные о той области экрана, где они находятся.

Вы сможете создавать с ребятами любые программы: расчетные, графические, обрабатывающие тексты; обучающие и даже обучаемые. Ваши ученики смогут изучать Лого все школьные годы, создавая, играя и работая с простыми картинками и мультиками, а позже с серьезными программами. А вы сможете вести уроки на Лого, начиная с младших классов и кончая углубленными курсами со старшеклассниками.

Одним словом, начните работать с Лого и вам нескоро понадобится еще что-то!

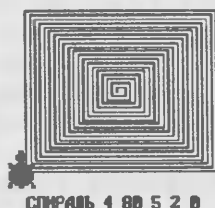
Вот несколько примеров программ.

Фирменный знак» графики Лого — спирали

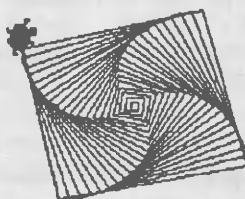
```
to СПИРАЛЬ :m :n :x :s :z
repeat :n [(fd :x rt (360 / :m) + :z
make "x :x + :s]
end
```

где m — определяет форму спирали (4 — квадратная, 3 — треугольная и т.д.);
 n — число проходов;
 x — длина прохода;
 s — на сколько увеличивается (уменьшается) каждый проход;
 z — угол «закручивания».

При выполнении с различными значениями параметров и результаты будут самые разнообразные. Например:



СПИРАЛЬ 4 80 5 2 0



СПИРАЛЬ 4 80 5 2 1



СПИРАЛЬ 160 300 0.01 0.01 0

Построение графиков функций

```
to ГРАФИК :f :m
cg setsh 10
ОСИ :m
make "x -150 / :m
repeat 300 [make "y :m * run :f
if and:y < 85 :y > -85
[pu setx :x* :m sety :y pd stamp]
make "x :x + 1 / :m]
end
```

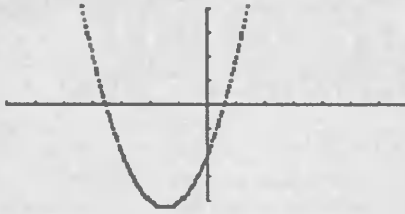


ГРАФИК [:x * :x + 3 * :x - 2] 20

(График функции $y = x^2 + 3x - 2$, увеличенный в 20 раз).

Моделирование движения

а) отражение мячика от стен:

```

to МЯЧ
РАМКА
setsh 12 home seth random 360
make "h heading
ДВИЖЕНИЕ_1
end

to ДВИЖЕНИЕ_1
if or y < -50 y > 50
  [seth 180 - :h make "h heading]
if or y < -100 y > 100
  [seth 0 - :h make "h heading]
fd 1 ДВИЖЕНИЕ_1
end

to РАМКА
cg pu setpos [-100 -50] pd
repeat 2
  [fd 100 rt 90 fd 200 rt 90] pu
end

```

Выполнение программы МЯЧ



б) моделирование движения в поле тяготения:

```

to СНАРЯД :vx :vy :g
РАМКА
setpos [-100 -50] setsh 12 pd
ДВИЖЕНИЕ_2
end

```

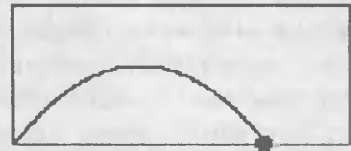
```

to ДВИЖЕНИЕ_2
setx x + :vx
sety y + :vy
make "vy :vy - :g
if (and x < 100 x > -100
    y < 50 y > -50) [ДВИЖЕНИЕ_2]
end

```

где vx и vy — скорости перемещения по горизонтали и по вертикали;
g — ускорение свободного падения.

Пример выполнения:



СНАРЯД 1 1.5 0.02

в) моделирование колебаний:

```

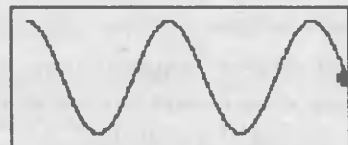
to МАЯТНИК :vx :vy :k
РАМКА
setpos [-90 40] setsh 12 pd
ДВИЖЕНИЕ_3
end

to ДВИЖЕНИЕ_3
setx x + :vx
sety y + :vy
make "vy :vy - :k * :y
if (and x < 100 x > -100
    y < 50 y > -50) [ДВИЖЕНИЕ_3]
end

```

где k — коэффициент упругости.

Пример выполнения:



МАЯТНИК 0.3 0 0.0005

Сортировка

Как мы вручную сортируем числовую последовательность (например, по возра-

станию)? Находим минимум, выписываем, вычеркиваем из исходных данных. Опять находим минимум, выписываем и т. д., пока список не кончится.

Нахождение минимального элемента и его номера в списке:

```
to МИН :l
make "m first :l make "i 1 make "n 1
repeat (count :l) - 1
  [make "i :i + 1
  if (first bf :l) < :m
    [make "m first bf :l make "n :i
    make "l bf :l]
op se :m :n
end
```

Исключение элемента из списка:

```
to ИСКЛ :n :l
make "q [ ]
repeat :n
  [make "q lput first :l :q make "l bf :l]
op se bl :q :l
end
```

Сортировка:

```
to СОРТ :l
make "r [ ]
repeat count :l
  [make "эл first МИН y :l
  make "r lput :эл :r
  make "l ИСКЛ last МИН :l :l]
op :r
end
```

Пример выполнения:

```
print СОРТ [1 366 -9 5 2 3 1]
```

```
-9 1 1 2 3 5 366
```

И никаких пузырьков!

Ханойская башня

```
to ХБ :n :a :b :c
ifelse :n = 1 [op word :a :b]
  [op (se ХБ :n - 1 :a :b :c
  word :a :b
  ХБ:n - 1 :c :b :a)]
end
```

где n - число дисков;

a - стержень, с которого снимаем;

b - стержень, на который перекладываем;
c - рабочий стержень.

Пример выполнения:

```
print ХБ 3 1 2 3
```

(т. е. надо переложить три диска с первого стержня на второй, используя в качестве вспомогательного третий).

```
12 13 23 12 31 32 12
```

(т. е. сначала переложить верхний диск с первого стержня на второй, затем - с первого на третий и т. д.).

Программы намеренно выбраны сложнее - для тех, кто считает, что Лого - только для малышей. На самом деле, Лого - прекрасное средство для развития мышления и самостоятельных исследований в самых разных интеллектуальных областях и с разными уровнями сложности.

Мне удалось вас убедить? Тогда предлагаю вам свою методику преподавания начал алгоритмизации и программирования в 8-9 классах, рассчитанную на 1-2 урока в неделю. Появилась она в результате четырех лет работы с Лого (преимущественно в 8-11 классах). Три года - на УКНЦ, а последний год - в классе IBM PC/AT (с системой LogoWriter, v2, в этой версии написаны все представленные программы). Литературы по Лого очень мало (у нас), поэтому пришлось самостоятельно разрабатывать методику и подбирать задачи.

Последовательность изучения тем:

- I. Графика черепашки; простые циклы-повторения.
- II. Выделение вспомогательных алгоритмов-процедур.
- III. Процедуры с переменными параметрами.
- IV. Вычислительные и диалоговые программы.
Циклы с переменными параметрами («N раз»).
- V. Разветвляющиеся алгоритмы, условные ветвления.
- VI. Рекурсия. Циклы типа «Пока».
- VII. Организация циклов в программах.

НАЧАЛА АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

8 класс (1 урок в неделю)

ИСПОЛНИТЕЛЬ ЧЕРЕПАШКА. ГРАФИЧЕСКИЕ КОМАНДЫ

ЗАНЯТИЕ 1

В вашем распоряжении - почти настоящий робот. Передвигаясь по экрану, черепашка оставляет след - рисунок. Команды управления черепашкой:

FORWARD число или **FD** число - вперед на указанное число шагов;

BACK число или **BK** число - назад на указанное число шагов;

RIGHT число или **RT** число - повернись вокруг своей оси на указанное число градусов по часовой стрелке;

LEFT число или **LT** число - повернись вокруг своей оси на указанное число градусов против часовой стрелки;

RG (reset graphics) - стереть рисунки и установить черепашку в исходное положение;

HOME - установка Черепашки в исходное положение.

Правила:

переключение шрифта РУС/ЛАТ - клавиша F5;

между командой и числом нужно оставлять пробел;

набрал команду - нажми ENTER.

Задания:

- нарисовать на экране квадрат;
- заштриховать экран «в косую линейку»;
- измерить длину и ширину экрана;
- нарисовать треугольник.

(подавляющее большинство ребят стараются изобразить треугольник, в котором один угол прямой, а остальные - по 45° и «конец» с «началом» не сходятся. Подсказка - нужно рисовать равносторонний).

ЗАНЯТИЕ 2

Задания:

1) нарисовать в тетрадях результат выполнения следующих последовательностей команд, обозначив схематически на чертежах начальное и конечное положение черепашки:

а) **FD 30 RT 30 FD 60**

б) **FD 40 LT 90 FD 40 LT 90 FD 40**

должно получиться так:



2) дополнить вторую последовательность команд так, чтобы получился полный квадрат.

Проверить получившееся на компьютере.

Переход от пошагового режима (ввод по одной команде) к выполнению последо-

вательности команд - программы. Команды отделяются друг от друга пробелом.

Если присмотреться к программе, рисующей квадрат, то можно увидеть, что в ней 4 раза повторяется пара команд:
FD 40 LT 90.

Новая команда, которая избавит нас от скучных повторов: REPEAT сколько раз [что повторять].

Например, чтобы нарисовать квадрат, можно дать команду:
REPEAT 4 [FD 40 LT 90]

Самостоятельная работа: поэкспериментировать с командой REPEAT.

ЗАНЯТИЕ 3

Приемы редактирования командной последовательности: как стереть ошибку и как вставить пропущенное.

Вопрос: можно ли в программе для рисования квадрата использовать другие команды? (Вместо FD можно взять BK, вместо LT - RT).

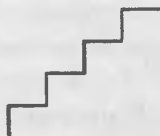
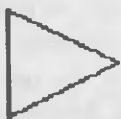
Задания:

1) нарисовать в тетрадах, что получится, если выполнить все возможные комбинации;



2) работая за компьютером проверить получившийся рисунок;

3) используя команду повтора, написать программы для рисования равно- стороннего треугольника и «лесенки».



ЗАНЯТИЕ 4

Разбор программы построения равно- стороннего треугольника. Внутренние углы - по 60° . Чтобы нарисовать треугольник и вернуться в исходное положение, чере- пашка должна 3 раза повернуться на внешний угол 120° .

(Например, REPEAT 3 [FD 60 RT 120]).

Черепашка может двигаться и без сле- да - это необходимо для получения на экране нескольких отдельных изображений.

Две новые команды:

PU (pen up) - поднять перо (после это- го черепашка двигается, не оставляя ри- сунка);

PD (pen down) - опустить перо (обрат- ная команда).

Задания:

написать и отладить программы для следующих рисунков:

ПУНКТИР



ДВА КВАДРАТА



ПИЛА



ДОМИК



ЗАНЯТИЕ 5

Правильные многоугольники

Какая геометрическая фигура должна быть следующей в этом ряду?



(правильный треугольник, правильный четырехугольник, ...)

Дальше должен быть правильный пятиугольник. А как написать для него программу?

`REPEAT 3 [FD 40 RT 120]` – правильный треугольник. Черепашка трижды повернулась на 120° и вернулась в исходную точку.

$3 \cdot 120 = 360$ – полный оборот.

`REPEAT 4 [FD 40 RT 90]` – правильный четырехугольник.

$4 \cdot 90 = 360$ – полный оборот.

Видимо, чтобы нарисовать пятиугольник, черепашка должна пять раз пройти нужное расстояние и повернуться на пятую часть полного оборота ($360 / 5 = 72$):

`REPEAT 5 [FD 40 RT 72]` – правильный пятиугольник.

А шестиугольник?

`REPEAT 6 [FD 40 RT 60]` – правильный шестиугольник.

А дальше – на что будет похож, скажем, правильный шестидесятиугольник? На окружность! И чем больше сторон, тем ближе к окружности. (Надо только не забыть уменьшать сторону, чтобы «уместиться» в пределах экрана).

Задания:

1) написать (в тетради) в общем виде программу для рисования правильного N -угольника со стороной A ;

(`REPEAT N [FD A RT 360 / N]`)

2) написать программу, рисующую «окружность»;

3) написать программу, рисующую следующую картинку:



ЗАНЯТИЕ 6

Решение задач:



а)



б)

ЗАНЯТИЕ 7

Процедуры

На предыдущем занятии пришлось несколько раз переписывать одну и ту же последовательность команд для рисования треугольника. Оказывается, в таком случае эту программу можно записать один раз (как в записную книжку) в специальном месте – «на обратной стороне листа», нажав клавиши `CTRL+F`. Записанной программе нужно дать имя и затем, сколько бы раз она ни потребовалась, достаточно вызвать ее по этому имени. Таким образом мы как бы обучаем черепашку новым командам. Этот процесс называется описанием процедур.

Правила описания процедуры:

- 1) придумать имя (из одной или нескольких букв или цифр);
- 2) «перевернуть лист» – нажать клавиши CTRL+F;
- 3) написать заголовок – специальное слово TO и имя;
- 4) написать саму программу;
- 5) написать слово END;
- 6) перевернуть лист - нажать клавиши CTRL+F.

На «обратной стороне листа» окажется описание – что должно выполняться, если в программе встретится новая команда. Все три части описания должны начинаться с новой строки. Имена процедур должны отличаться от команд и от имен других процедур.

Задания:

- 1) описать процедуру рисования правильного треугольника;
- 2) используя эту новую команду, написать программы, рисующие следующие картинки:



а)



б)



в)

ЗАНЯТИЕ 8

На предыдущем занятии решалась задача - цветок с четырьмя треугольными лепестками.

Описание лепестка:

```
TO TR
REPEAT 3 [FD 40 RT 120]
END
```

Программа рисования цветка:

```
REPEAT 4 [TR RT 90]
```

Задания:

- 1) написать процедуры, рисующие квадратный, пятиугольный, круглый лепесток;
- 2) написать программы, рисующие цветы с разным количеством и различной формой лепестков.

ЗАНЯТИЕ 9

Команды работы с цветом

До сих пор черепашка рисовала белым по черному. В нашем распоряжении еще 14 цветов - всего 16. Они перенумерованы: с 0 по 15. Новые команды:

SETBG число (set background) – установка цвета фона;

SETC число (set color) – установка цвета черепашки и ее пера;

FILL - закрасить замкнутый контур, в котором находится черепашка, ее цветом. При этом внутрь контура черепашка должна попасть без следа, после чего перо должно быть опущено.

Задания:

- 1) нарисовать несколько разноцветных геометрических фигур;
- 2) нарисовать и закрасить окружность;

3) в полученном круге нарисовать треугольник и закрасить в другой цвет.

ЗАНЯТИЕ 10

Шахматная доска

Шахматная доска - 64 квадратные клеточки, половина из которых закрашена. При решении этой уже достаточно сложной задачи поможет разбиение ее на подзадачи и описание более простых частей как процедур.

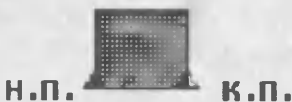
Первая процедура - просто квадратик. Пусть мы нарисуем его по часовой стрелке. Но остановиться черепашка должна уже не в исходном положении, а там, откуда она начнет рисовать следующий, закрашенный квадратик.



```
(TO SQ_1
REPEAT 4 [FD 15 RT 90]
RT 90 FD 15 LT 90
END)
```

Чтобы получить закрашенную клетку, можно использовать готовую процедуру (а не начинать сначала). Это уже получится процедура второго уровня, т.е. использующая ранее описанную процедуру.

Тогда можно будет, чередуя вызовы первой и второй процедур, получить первую строчку (из восьми квадратиков). Для этого во второй процедуре черепашку опять нужно оставить там, откуда она начнет рисовать следующий, незакрашенный квадратик.



```
(TO SQ_2
SQ_1 LT 45 PU FD 5
PD FILL BK 5 RT 45
END)
```

Общее полезное правило: конечное положение черепашки в процедуре должно быть начальным положением для следующей процедуры.

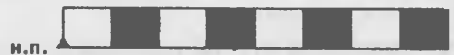
Задание:

написать процедуры для закрашенного и незакрашенного квадратиков и получить из них строчку.

ЗАНЯТИЕ 11

Шахматная доска (продолжение)

На прошлом занятии была получена программа, чередующая восемь клеточек (REPEAT 4 [SQ_1 SQ_2]). Вся шахматная доска - восемь строчек, 4 нечетные и 4 четные (в них клетки чередуются наоборот). Значит, нужно написать еще две процедуры - строчка нечетная и строчка четная (не забывая о правиле).



```
к.п. ▲
(TO LINE_1
REPEAT 4 [SQ_1 SQ_2] PU LT 90 FD
15 * 8 RT 90 BK 15 PD
END)
```

ЗАНЯТИЕ 12

Шахматная доска (окончание)

Полученные процедуры для строчек относятся к третьему уровню: в них используется процедура, которая сама использует процедуру.

И наконец, последняя, «верхняя» процедура - чередование нечетных и четных строк.

```
(TO A
REPEAT 4 [LINE_1 LINE_2]
END)
```

Вся пирамида процедур по возможности должна быть написана ребятами самостоятельно. Программы могут и должны различаться в деталях и по плану решения (например, доску можно строить не по строчкам, а по столбцам, или рисовать все строчки не слева направо, а челноком). Принципиально то, что общая программа собирается из процедур, как дом из блоков. А блоки, в свою очередь, могут состоять из более мелких деталей и т. д.

ЗАНЯТИЯ 13-14

Вопрос: как получить полуокружность?

Нарисовать только половину сторон соответствующего многоугольника (например, REPEAT 180 [FD 1 RT 1]).

Каким числом регулируется величина этой полуокружности? Очевидно, размером стороны многоугольника.

Задание:

описать необходимые процедуры рисования полуокружностей и, используя их, написать программы для следующих картинок:

ВОЛНЫ



ПРУЖИНКА



ПЕРО



ЦВЕТОК



ЗАНЯТИЕ 15

Процедуры с переменными параметрами

Команды языка LOGO можно разделить на две группы.

Одни команды выполняются всегда одинаково и после команды никакой дополнительной информации указывать не нужно, а другие требуют указания числовых или иных данных.

Игра - разминка и повторение:

на доске - два списка; ребята называют известные им уже команды первого типа, затем - второго. Побеждает (получает пятерку) тот, кто назовет последнюю команду в каждом списке.

(I - HOME, RG, PU, PD, TO, END, FILL;

II - FD ?, BK ?, RT ?, LT ?, SETC ?, REPEAT ? [?] и т. д.)

Но этот список можно расширять, придумывать новые команды (описание процедур).

Предположим, мы описали простейшую процедуру:

```
TO A
REPEAT 4 [ FD 60 RT 90 ]
END
```

К какой группе относится новая команда A?

Правильно, к первой - чтобы ее выполнить, достаточно напечатать ее имя и нажать ввод, она всегда выполняется одинаково.

К первой группе относились все процедуры, которые мы писали до этого. Оказывается, можно изменить процедуру A так, чтобы она принадлежала ко второй группе (например, мы сможем получать квадраты разного размера). Для этого надо:

1) то число, которое мы хотим сделать изменяемым (переменный параметр) заменить на имя (букву, несколько букв) со специальной меткой - ":".

2) указать это имя (с двоеточием) в заголовке процедуры:

```
TO A :Z
REPEAT 4 [ FD :Z RT 90]
END
```

Теперь новая команда А относится ко второй группе и ее можно вызывать только с указанием числа. Например, А 10 - квадрат со стороной 10.

Задание:

КВАДРАТЫ



БАБОЧКА



(процедура, рисующая окружность переменного размера направо; процедура, рисующая окружность переменного размера налево; общая процедура – весь рисунок).

ЗАНЯТИЕ 16

Используя описание процедур с переменным параметром и правило о конечном положении черепашки, решить следующие задачи:

КВАДРАТЫ



ПИРАМИДА



СНЕГОВИК



ЗАНЯТИЕ 17

Отрицательное значение аргумента

Некоторые команды LOGO «понимают» отрицательные числа. Так, $FD - 90 = BK 90$ (и наоборот); $RT - 90 = LT 90$ (и наоборот).

Это свойство можно использовать, чтобы сделать программы более короткими и простыми (в основном, симметричные рисунки).

Задание:

получить следующие картинки, используя только одну вспомогательную процедуру с положительными и отрицательными аргументами

- 1) бабочка (см. занятие 15);
- 2) бантик

БАНТИК



```
(TO TR :x
REPEAT 3 [ FD :x RT 120 ]
END
TO бантик
RT 30 TR 50 TR -50
END)
```

ЗАНЯТИЕ 18

Процедуры с несколькими переменными параметрами

Переменным параметром может быть не только размер!

Сколько чисел можно сделать переменным параметром в процедуре, рисующей правильный многоугольник?

(Например,

```
TO P
REPEAT 18 [ FD 6 LT 20 ]
END)
```

Нет, не три, а только два.

И если одно из них определяет размер, то второе — форму!

Процедура «Универсальный многоугольник»

```
TO A :N :X
REPEAT :N [ FD :X RT 360 / :N ]
END
```

Если второй переменный параметр определяет размер (это уже привычно), то первый — ФОРМУ.

Задание:

используя только одну вспомогательную процедуру получить:

- 1) многоугольники разного размера и формы;
- 2) картинку б) из занятия 6;
- 3) «цветы» с различным количеством и формой лепестков.

ЗАНЯТИЕ 19

Команды ввода-вывода PRINT и INSERT

выводят на экран не рисунки и перемещения черепашки, как знакомые уже графические команды, а:

- 1) различные сообщения — для этого надо их заключать в [];
- 2) результаты вычислений — действий над числами и значениями переменных, причем записывать арифметические выражения надо в строчку, деление обозначается "/", а не ":", умножение — "*",

скобки только круглые, даже если одни в других.

Задание:

написать программу (в виде процедуры с переменными параметрами), которая выводит:

- а) среднее арифметическое двух чисел;
- б) площадь стен комнаты, если известны ее длина, ширина и высота;
- в) печатает 10 раз "Ура!";
- г) заданное число раз фразу "Ученье свет, а неученье — тьма!";
- д) заданное число раз заданное слово.

```
(TO A :X :Y
REPEAT :X [ PR :Y ]
END).
```

Последнюю задачу почти никто не выполняет самостоятельно, что позволяет привлечь внимание к тому, что переменная может быть не только числом (А 17 [Маша ела кашу]).

ЗАНЯТИЯ 20-21

Команда присваивания MAKE

Задача: написать программу, которая выводит последовательность целых чисел: 1 2 3 4 5 6 Самостоятельные попытки доводят ребят только до использования команд REPEAT и PR :N. Новая команда MAKE позволяет обновлять значение переменной в цикле и устанавливать ее первоначальное значение:

```
TO A
MAKE "N 1
REPEAT 100 [ PR :N
MAKE "N :N + 1 ]
END
```

Задание:

написать программу (в виде процедуры с переменным параметром), которая:

- а) выводит первые 100 нечетных чисел;
- б) выводит все положительные четные, начиная с 200, в обратном порядке;
- в) рисует квадратную спираль;

```
(TO A :N
MAKE "N 5
REPEAT 30 [FD :N RT 90
MAKE "N :N + 5]
END).
```

- г) рисует треугольную спираль;
- д) проигрывает «звуковую дорожку»

```
(TO A :N
MAKE "N 80
REPEAT 50 [TONE :N 1
MAKE "N :N + 4]
END).
```

Дополнительное задание: изменить угол поворота в в) и г) на 1-2 град.

е) подсчитывает потомство одной амебы за двое суток, если каждая амеба через два часа делится на две.

ЗАНЯТИЕ 22

Легенда о происхождении шахмат: мудрец попросил у раджи плату: за первую клетку доски - 1 рисовое зерно, за каждую следующую - вдвое больше, чем за предыдущую.

Задание:

написать программу, выводящую число зерен, которое нужно уплатить за каждую из 64 клеток. (Программа «работает»

лишь до сороковой клетки - дальше числа слишком велики).

ЗАНЯТИЕ 23

Разбор задачи о накоплении суммы - в цикле изменяются уже 2 переменные.

Задание:

написать программу:

- а) подсчитывающую сумму первых 100 натуральных чисел;
- б) подсчитывающую количество зерен, которое раджа был должен заплатить за первые 32 клетки шахматной доски;
- в) возводящую заданное число в заданную (натуральную) степень.

```
(TO A :X :N
MAKE "Z :X
REPEAT :N [ MAKE "Z :Z * :X]
[INSERT :X [В СТЕПЕНИ ] :N
[PABHO] :Z]
END)
```

ЗАНЯТИЕ 24

Команды ввода **READCHARACTER** и **READLIST** вместе с командами вывода позволяют программировать диалог с компьютером. Их действие - заставить компьютер ожидать ввода с клавиатуры. Особенность - надо сразу указывать, что делать с полученным с клавиатуры символом или последовательностью (списком). А что с ними делать? Запомнить, с помощью команды **MAKE**.

Демонстрация, объяснение действия команды **FIRST** (есть команда для ввода одного символа и списка слов или чисел,

а для ввода одного слова или числа специальной команды нет - используется FIRST READLIST).

Разбор программы по сценарию:

вопрос: "Как тебя зовут?"

ответ: "****"

вывод сообщения

"Очень приятно, ****"

(TO A

PR [КАК ТЕБЯ ЗОВУТ?]

MAKE "L FIRST READLIST

TYPE [ОЧЕНЬ ПРИЯТНО,] PR :L

END)

Задание:

написать программу, которая:

а) спрашивает, сколько вам лет и, получив ответ, называет год вашего рождения;

б) спрашивает, какой у вас рост и, получив ответ, называет число, которое не должен превышать ваш вес (рост (см) - 110).

(Продолжение следует).

ВНИМАНИЮ ПРЕДПРИНИМАТЕЛЕЙ И РУКОВОДИТЕЛЕЙ ПРЕДПРИЯТИЙ И ОРГАНИЗАЦИЙ!

Редакция принимает заявки на публикацию объявлений и рекламных материалов в журнале «Финансы» и в приложении к журналу «Нормативные акты по финансам, налогам и страхованию».

Размещение рекламы в наших изданиях позволит расширить сферу Вашего влияния и увеличить объем сбыта продукции и услуг.

Если Вас заинтересовало наше предложение, в адрес редакции следует направить рекламный текст с гарантийным письмом или копией платежного поручения.

По всем вопросам, связанным с размещением рекламы, необходимо обращаться по адресу:

103050, Москва, ул. Тверская, 226. Редакция журнала «Финансы».

Телефон для справок: 299-96-16.

**ВАША РЕКЛАМА В ЖУРНАЛЕ «ФИНАНСЫ» БУДЕТ АДРЕСОВАНА ИМЕННО ТЕМ
ЛЮДЯМ, КОТОРЫЕ ДЕЙСТВИТЕЛЬНО ОБЛАДАЮТ ВОЗМОЖНОСТЬЮ И ПРАВОМ
ПРИНЯТИЯ ФИНАНСОВЫХ РЕШЕНИЙ!**

В РЕДАКЦИИ ЖУРНАЛА «ФИНАНСЫ» МОЖНО ПРИОБРЕСТИ СЛЕДУЮЩИЕ ИЗДАНИЯ:

1. Страхование (бухгалтерский учет, налогообложение, лицензирование). Специальное приложение к журналу «Финансы». 1993. — Вып. 1
2. Налоги Российской Федерации. Специальное приложение к журналу «Финансы». Часть 4. 1993.
3. Министерству Финансов России 90 лет. Специальное приложение к журналу «Финансы». 1993.
4. Корреспонденция счетов. 2000 основных хозяйственных операций предприятия. Библиотека журнала «Бухгалтерский учет». Вып. 1 и 2. М.: Бухгалтерский учет, 1992.

ФИРМА «КОРВЕТ»

ПРЕДЛАГАЕТ

поставка компьютерных классов на базе IBM PC
поставка компьютерных классов УКНЦ-01.01 (БК-0011М)
установка и запуск ваших компьютерных классов
модернизация компьютерных классов УКНЦ-01 в УКНЦ-01.01
техническое обслуживание средств вычислительной техники
поставка комплекта программного обеспечения для классов УКНЦ-01.01 (БК-0011М)
разработка программного обеспечения по индивидуальному заказу
ремонт мониторов, дисководов, блоков питания, ПЭВМ БК-0010, БК-0011, БК-0011М
установка и запуск лингафонных кабинетов

Обращайтесь к нам по адресу:

142500, Московская область, г.Павловский Посад, ул.Ленина, д.44 (с 10 до 18 ч.).
Телефон: (243) 2-01-62 — для Москвы и Московской области;
(09643) 2-01-62 — для других районов.
Телефон для справок: (095) 134-53-88.
Адрес для переписки: 142500, Московская область, г.Павловский посад, узел связи, а/я 111. ТОО «КОРВЕТ»



НИИЯФ МГУ им. М.В.Ломоносова отдел микроэлектроники предлагает:

Расширение возможностей "Корвета":

- замена процессора на Z-80;
- увеличение оперативной памяти;
- подключение электронного диска;
- подключение винчестера.
- консультации и ремонт;
- комплект диагностического оборудования для ремонта;
- комплект технической документации;
- продажа системных блоков, дисководов, мониторов, джойстиков, "мышек", защитных экранов.

Новые версии программ для "Корвета"

- операционная система;
- редактор текстов;
- сервисные программы для работы с дисками.
- русификатор для IBM PC, поддерживающий кодировки IBM PC и "Корвета";

- программы для IBM PC, поддерживающие обмен PC — "Корвет";
- игровые программы.

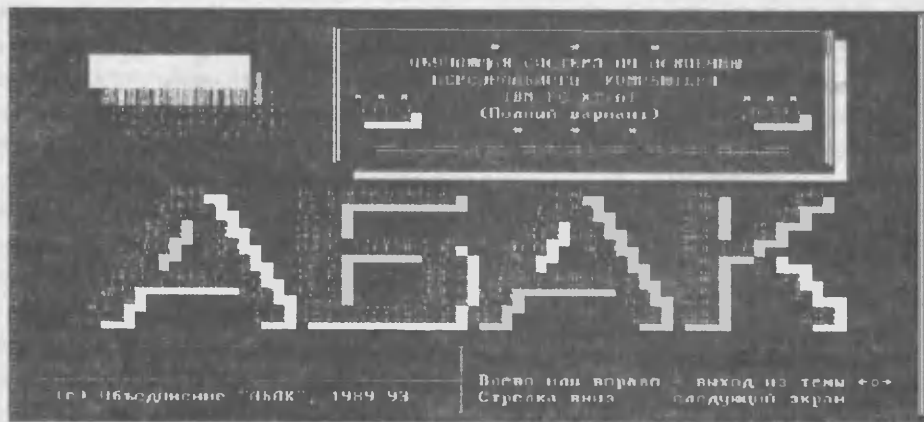
Книги:

- методическое пособие по dBase II для "Корвета";
- "Знакомьтесь: Корвет".

Для владельцев IBM PC:

- Впервые полный школьный курс "Механика Ньютона" — методическое пособие и лабораторный практикум, разработанный преподавателями физического факультета МГУ с учетом специфики изложения механики в школе.

С заказами обращайтесь по адресу: 119899, ГСП, Москва, Воробьевы горы, МГУ, НИИЯФ, "МикС", тел. (095) 939-49-49



Н.П. Пиликов, Б.Н. Болотин,
Н.Т. Казулин, Д.Э. Соловьев

РАБОТАТЬ НА КОМПЬЮТЕРЕ НАУЧИТ КОМПЬЮТЕР!

Может ли компьютер учить?

Начиная с 70-х годов у нас в стране стала развиваться такая отрасль научно-технических знаний, как «Разработка компьютерных обучающих систем». Вначале это направление сулило просто фантастические результаты. Действительно, возможности компьютера по скорости и объему обрабатываемой информации намного превышали возможности человека, и «горячие головы» стали всерьез поговаривать об отмирании профессии учителя. Но уже через 5 – 10 лет работы восторженный оптимизм по этому поводу сменился рассудительным реализмом. Одной из главных причин неудач, связанных с заменой учителя компьютером (замечу, не у нас, а в таких развитых странах, как США), сторонники компьютерного обучения называли недостаточность парка компьютеров. С высоты же 1993 года можно,

не сильно ошибаясь, сказать, что дерзкая идея по замене учителя компьютером потерпела полное фиаско.

И тем не менее что-то было наработано за эти годы. Коснемся тезиса о нехватке компьютеров. В некоторых местах (как правило, в университетах) были созданы просто заповедные условия для компьютерного обучения. На десятках мощнейших по тем временам ЭВМ выполнялись гигантские обучающие программы, созданные крупными коллективами программистов и энтузиастами компьютерного обучения. На студентов обрушивались десятки часов крайне дорогостоящего машинного времени, и тем не менее ожидаемого фантастического эффекта получено не было. Стало быть, принципиальное решение проблемы лежит не в области объемов производства компьютеров. Выяснилось, что большинство обучающих систем не только не приносят обучаемым ре-

альной пользы, но и настраивают их против идеи компьютерного обучения. По этому вопросу можно было бы написать толстую книгу, пытаясь его подробно проанализировать, но ограничимся выводом. Прежде чем браться за разработку обучающей системы, нужно ответить на четыре взаимосвязанных вопроса.

- *Кого учить?*
- *Чему учить?*
- *Как учить?*
- *Кто будет наполнять оболочку обучающей системы знаниями?*

Подавляющее большинство обучающих систем было ориентировано на школьников или студентов высших учебных заведений, и учить они могли всему (это следовало из заявлений авторов разработок). Кроме того, разработчики обучающих систем представляли пользователям только оболочку — инструмент. А далее каждому желающему предлагалось наполнить эту оболочку знаниями. Под желающими они, очевидно, имели в виду преподавателей конкретных дисциплин (предметников), которые, не обладая достаточными знаниями в области программирования, но имея большой багаж знаний в своей области, смогут профессионально создать наполнение. Но, как свидетельствует мировой опыт разработки обучающих систем, таких желающих оказалось немного. Объясняется это не тем, что преподаватели очень ленивы или ретрограды, а тем, что, несмотря на кажущуюся простоту этого занятия, оно требует не меньшего опыта и профессионализма, чем преподавание традиционным классическим методом. И самое главное то, что для разработки хорошего наполнения обучающей системы нужно поступиться своим личным временем и при этом вложить колоссальное количество труда (интеллектуальных усилий и времени), как правило, не оплачиваемого.

Проанализировав несколько десятков обучающих систем, в том числе такой авторитетной фирмы, как «Microsoft Corp.»,

коллектив сотрудников объединения «АБАК» сделал вывод о причинах неудач, связанных с разработкой действительно нужных пользователям компьютерных программ данного назначения. Обучающая система должна быть рассчитана на конкретную группу пользователей, должна содержать в себе все лучшее, что программистами и методистами разных стран по крупицам удалось собрать и придумать, а главное — это должен быть законченный продукт, начиная с оболочки и кончая наполнением, дополненный необходимой методической литературой.

Далее речь пойдет об обучающей системе по освоению персонального компьютера IBM PC XT/AT «АБАК» версии 6.2, являющейся разработкой этого объединения.

Кого учить?

В последние три года в России произошли очень благоприятные для разработчиков коммерческих программ перемены в области компьютеризации общества. Эти перемены касаются прежде всего резкого увеличения парка персональных компьютеров вследствие их существенного удешевления.

Действительно, если в 1989 году стоимость одного компьютера IBM PC AT-286 равнялась 8 тоннам хорошей колбасы, то летом 1993 года стоимость того же компьютера измерялась уже только 100 — 140 килограммами. То есть, персональные компьютеры подешевели примерно в 15 раз.

Последствия не заставили себя долго ждать. Если раньше преимущественными обладателями персоналок были только крупные государственные предприятия, причем работали на них только квалифицированные программисты, то теперь персональным компьютером располагает любое мало-мальски устойчивое в финансовом отношении малое предприятие. Мало того, нередко персональные компьютеры являются собственностью частных лиц.

Это обстоятельство породило достаточно серьезную проблему, связанную с эффективным использованием имеющегося парка персональных компьютеров. Раньше этой проблемы не было, поскольку персональные компьютеры попадали в руки только очень квалифицированных программистов, которые могли выжать из них все, что те могли дать, и, в частности, их загрузка составляла не менее 15 часов в сутки. Теперь же персоналки, во-первых, попадают в руки к новичкам и, во-вторых, эти новички покупают их для совершенно иных целей.

Если профи используют персональный компьютер как инструмент при разработке новых программ, то новичок покупает его для того, чтобы пользоваться уже готовыми программами и экономить свое рабочее время. Большинству из новичков персональный компьютер нужен для автоматизации бухгалтерии и ведения делопроизводства (подготовка различных писем, договоров и тому подобное). Но сможет ли новичок немедленно после покупки компьютера использовать его для достижения своих целей?

Ответ на этот вопрос не является однозначным.

С одной стороны, когда студента на экзамене спрашивают: «Что такое персональный компьютер?», то он должен перечислить пять-шесть признаков, в силу которых данную ЭВМ можно назвать персональным компьютером, причем важнейшим из этих признаков является возможность использования этой ЭВМ абсолютно неподготовленными людьми. Таким образом выходит, что на персональном компьютере не нужно учиться работать просто по определению.

С другой стороны, повседневная практика показывает, что это утверждение не может быть признано абсолютно верным. В самом деле, даже такое невероятно простое по сравнению с персональным компьютером устройство, как пишущая ма-

шинка, требует определенных навыков и знаний при обращении с ней.

Если мы попытаемся очертить круг минимальных знаний, которыми должен располагать пользователь персонального компьютера (то есть некий джентльменский набор), то получится как минимум книжка объемом 100 – 200 страниц. Здесь еще и еще раз следует подчеркнуть, что под пользователем мы имеем в виду не профессионала, объем знаний которого можно измерять десятками книг такого объема, а новичка, не собирающегося становиться профессионалом и желающего использовать компьютер только как средство, увеличивающее производительность его труда. Каким же образом получить новичку эти знания? Можно пойти учиться на курсы по освоению персонального компьютера, можно купить кучу книг и начать их штудировать, можно бесконечно обращаться за консультациями к своим знакомым, можно, наконец, взять преподавателя.

Но для того ли был куплен персональный компьютер, чтобы постоянно учиться на нем работать? Очевидно, нет.

И вот здесь нам остается, как волшебнику из сказки, произнести магическое заклинание – «Работать на компьютере научит компьютер». Конечно же, эту фразу не стоит понимать буквально, поскольку компьютер состоит из двух частей – это его аппаратура и программы, которые могут быть выполнены на этой аппаратуре. Стало быть, учит все же не компьютер, а обучающая система, выполненная на его базе. Кроме того, нельзя объять необъятное и научить буквально всему, но вот тому джентльменскому набору знаний, позволяющему использовать компьютер в качестве инструмента для экономии времени – можно. И сегодня это уже осуществляется.

Для кого же делалась обучающая система АБАК? Отнюдь не для профессионалов, желающих освоить еще несколько

АБАК™



(c) Объединение "АБАК", 1989-93

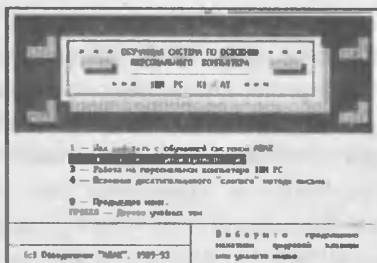
р/с 466702 в Акционерном
«Элексбанке» корр/с 285161000 в
РКЦ ГУ ЦБ России, г. Москва,
МФО 201791,
восьмиразрядный код 44583285

Работайте на компьютере научит компьютер!

если Вы оснастите свой компьютер нашей
обучающей системой «АБАК» версии 6.2

Чему учит обучающая система?

- * Как работать с обучающей системой
- * Основы компьютерной грамотности
 - компьютер IBM PC и устройства, входящие в его состав
 - основные понятия операционной системы MS-DOS
- * Работа на персональном компьютере IBM PC
 - оболочка DOS — Norton Commander v. 3.0
 - некоторые команды операционной системы MS-DOS v. 5.00
 - многооконный текстовый процессор ЛЕКСИКОН вер. 8.95
- * Освоение десятипальцевого «слепого» метода письма
 - Всего около 210 учебных тем.



Как учит обучающая система?

Вы выбираете любую тему обучения с помощью меню, и программа выводит на экран дисплея обучающий материал, состоящий из нескольких «страниц» текста. Когда вы их прочтете, обучающая система задаст вам несколько вопросов. После ответа на вопрос система «АБАК» подробно объяснит вам, в чем вы ошиблись, или, если вы ответили верно, даст дополнительный, более сложный материал. Такая методика обучения обеспечит глубокое и полное понимание изученного.

Вы освоите «слепого» десятипальцевый метод письма на клавиатуре компьютера с помощью 1300 упражнений из прекрасного учебника по машинписанию, которым придан «компьютерный блеск». Умение работать с клавиатурой компьютера со скоростью профессиональной машинистки необходимо квалифицированным пользователям.

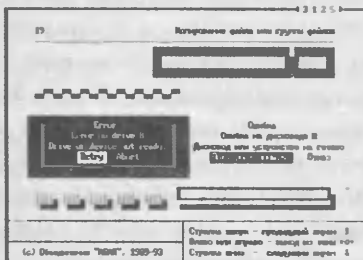
Для кого предназначена обучающая система?

Наши покупатели — руководители, бизнесмены, бухгалтеры, секретари — словом, все те, кто начинает или продолжает освоение персонального компьютера.

Сколько стоит обучающая система?

Ознакомительный вариант (первые 35% учебного материала) — 5 долларов США, полный вариант — 60 долларов. Оплата в рублях по курсу ММВБ на день оплаты. При оплате наличными — скидка 30%. При покупке 5-и или более установок — скидка 40%.

Нашу систему можно приобрести по почте. Присылайте письмо с копией платежного поручения и Вашим точным почтовым адресом и телефоном.



«АБАК» - лучшая в России обучающая система по освоению персонального компьютера!

189428, Москва, а/я 33, объединение «АБАК», т. (095)178-87-22

языков программирования или какую-нибудь специализированную и сложную прикладную программу. Она делалась для людей, которые пока еще не знают, как подойти к компьютеру, какую кнопку нажать и что из этого получится, то есть для самых начинающих. Она делалась для людей, которые «по уши» вязнут в своей основной работе и приобрели персональный компьютер для того, чтобы преуспеть именно в ней, а не в программировании. Она делалась для людей, которым некогда читать тома скучных справочников на компьютерную тематику. Она делалась для тех, кому некогда регулярно ходить на курсы по освоению компьютеров или встречаться с преподавателем. Ну и, конечно же, для тех, кто может позволить себе истратить еще пару процентов от стоимости своего персонального компьютера на оснащение его обучающей системой.

Чему учить?

Так что же это за джентльменский набор знаний, который является необходимым и достаточным минимумом для использования персонального компьютера в качестве своего незаменимого помощника? К счастью, ко времени разработки обучающей системы АБАК этот набор был достаточно точно определен во многих и многих пособиях и учебниках по персональным компьютерам. Нам оставалось лишь воспользоваться опытом других.

Сначала обучаемого учат, как работать с обучающей системой. Далее идет раздел «Основы компьютерной грамотности», состоящий из двух подразделов: «Устройство персонального компьютера IBM PC XT/AT» и «Основные понятия операционной системы MS-DOS». Изучение первого из них позволяет обучаемому узнать об основных частях компьютера и о том, как следует обращаться с его клавиатурой. Изучение второго позволяет обу-

чаемому принимать участие в беседах с профессиональными программистами и подготовленными пользователями.

Следующий раздел «Работа на персональном компьютере» состоит из трех больших подразделов: «Оболочка DOS – Norton Commander v. 3.0», «Некоторые команды операционной системы MS-DOS v. 5.0» и «Многооконный текстовый процессор ЛЕКСИКОН».

Здесь обозначены лишь укрупненные названия изучаемого материала. Весь этот материал разбит на достаточно мелкие учебные темы (каким именно образом организована обучающая информация, ниже будет описано довольно подробно). Всего этих учебных тем около двухсот.

Таким образом, проработав весь материал, который предлагает обучающая система АБАК, человек превращается из новичка, знающего только то, что без компьютера ему не обойтись, в грамотного пользователя.

Для желающих освоить слепой десятипальцевый метод письма имеется раздел, позволяющий после его проработки печатать со скоростью и качеством опытной машинистки.

Как учить?

Обсудим здесь очень коротко некоторые принципиальные положения.

Процесс обучения имеет двусторонний характер. С одной стороны – средство обучения, а с другой – обучаемый. Что в данном случае подразумевается под обучаемым, понятно. Это человек, желающий освоить работу на персональном компьютере. Под средством обучения будем иметь в виду:

- *учителя (преподавателя);*
- *книгу (учебник);*
- *компьютер (обучающую компьютерную систему).*

Попробуем выделить функции средства обучения и функции обучаемого в процессе обучения. Итак.

Функции средства обучения:

- *объяснение учебного материала;*
- *ответы на вопросы обучаемого;*
- *постановка вопросов обучаемому, анализ ответов обучаемого и реакция на его ответы;*
- *показ обучаемому, как нужно выполнять (имеется в виду то, чему учишь).*

Функции обучаемого:

- *усвоение учебного материала;*
- *постановка вопросов средству обучения;*
- *ответы на вопросы средства обучения;*
- *тренировка в предмете обучения.*

Теперь попробуем сравнить средства обучения (учитель, книга, компьютер) с точки зрения как самого средства обучения, так и обучаемого.

Пункт первый – объяснение материала. Несомненно, что есть большое количество учителей (преподавателей), способных с помощью данного им природой таланта, эмоциональности, наконец, человеческого обаяния сделать процесс объяснения материала увлекательным, живым, интересным и доходчивым. Но в целом приходится признать, что нет лучше способа ознакомления с материалом, чем работа с книгой (учебником).

- *Книга как средство обучения одинаково доступна всем, а талантливых учителей не так уж много.*
- *Книгу можно читать в том темпе, который устраивает обучаемого. В аудитории же темп подачи информации задан один на всех, что приводит одних в скуку, поскольку этот темп для них слишком медленный, а других в отчаяние потому, что они не успевают за мыслью преподавателя.*
- *Книга в отличие от учителя никогда не ошибается, не устаёт, не может оби-*

деть (назвав бестолковым), не повышает голоса.

- *К одному и тому же учебному материалу, содержащемуся в книге, можно обращаться сколько угодно много раз.*
- *Книга более физиологична. Ее читать гораздо удобнее, чем слушать учителя (разумеется, за исключением тех случаев, когда приходится иметь дело с талантливым учителем).*

Таким образом, книга как средство обучения далеко оставляет позади учителя в плане объяснения учебного материала.

Теперь обратимся к компьютеру как к средству объяснения учебного материала. На первый взгляд все достоинства книги почти автоматически переносятся на компьютерную систему обучения, если бы не два «но». Дисплей как средство восприятия информации (попросту чтения текста) крайне нефизиологичен. Глаза при чтении текста с дисплея, по сравнению с чтением книги, сильно напряжены и устают. Положение тела и, в частности, головы неудобно. Второе «но»: стоимость компьютера с его обучающей системой не соизмерима со стоимостью книги.

Таким образом, применение компьютера просто как средства объяснения материала в лучшем случае неоправданно.

Обратимся теперь к следующим перечисленным выше функциям, это ответы на вопросы (функция средства обучения) и постановка вопросов средству обучения (функция обучаемого).

Если постановка вопросов средству обучения (соответственно ответы на вопросы обучаемого) понимать не в некотором переносном смысле, а в прямом, то книге их задать нельзя в принципе (соответственно книга не может отвечать на вопросы ученика). Говоря о компьютере, можно было бы, конечно, порассуждать о системах искусственного интеллекта, экспертных системах и тому подобном, но сегодня, пожалуй, честнее признать, что в этом плане у учителя конкуренты отсутствуют.

Обсудим теперь две следующие функции: «Постановка вопросов обучаемому, анализ ответов обучаемого и реакция на его ответы» (функция средства обучения) и «Ответы на вопросы средства обучения» (функция обучаемого). Сравнивая учителя, книгу и компьютер, быстрее всего в данном случае можно расправиться с книгой, поскольку ничего другого, кроме вопросов в никуда и ни к кому, типа «Как вы считаете ...» или «Подумайте ...», книга в принципе задать не может. Об анализе и реакции на ответ обучаемого говорить также не приходится. Итак, будем сравнивать учителя и компьютер. Прежде всего обратим внимание на то, что по сравнению с компьютером учитель хоть и является более дешевым средством обучения (по крайней мере, в России), но разница уже не такая большая, как при сравнении компьютера с книгой, а в развитых странах это сравнение окажется в пользу компьютера.

Перейдем непосредственно к сравнению.

Во-первых, поскольку обучаемый работает с компьютерной обучающей системой, то задавание вопросов ему гарантировано, поскольку он работает с ней один на один. Когда же преподаватель задает вопрос аудитории (а так чаще всего и бывает), то после того, как кто-нибудь из обучаемых первым правильно ответит на этот вопрос, остальным уже делать нечего. И некоторые ученики, обладая «достаточной скромностью», вполне могут за весь учебный период ни разу не попытаться ответить на вопросы, задаваемые учителем. В том случае, если вопрос будет адресован конкретному обучаемому, то остальные обучаемые из данной аудитории вполне могут не утруждать себя размышлениями. Реакция же учителя на ответ обучаемого (как в случае правильного ответа, так и в случае неправильного) будет также адресована кому-то одному.

Остается теперь только поставить вопрос: «А может ли вообще учитель задавать вопросы?», имея в виду данный контекст. Оставляя право поразмышлять над ответом на этот вопрос читателю, мы убеждены, что в плане индивидуализации (персонализации) обучения компьютер несравненно превосходит учителя.

Во-вторых, заранее известно, что при подготовке (придумывании) вопросов обучающей системы количество интеллектуального труда, вложенного разработчиками этой системы, будет несравненно больше, чем труда учителя, хотя бы потому, что у разработчиков обучающей системы больше времени на подготовку этих вопросов. Отсюда и качество вопросов должно быть выше.

В-третьих, времени обучаемому на обдумывание ответа на вопрос компьютер предоставляет ровно столько, сколько ему для этого необходимо.

В-четвертых, компьютер не станет сердиться, в любых ситуациях будет корректен и дружелюбен, он бесконечно терпелив.

В-пятых, при неправильном ответе обучаемого компьютер отреагирует на этот ответ заранее тщательно продуманным образом.

В-шестых, в случае компьютерного обучения к одному и тому же вопросу при неудачных попытках ответа можно обращаться неоднократно.

В-седьмых, компьютер предоставляет несравненно более изысканные возможности по самому виду (типу, конструкции) вопросов.

В-восьмых, обучаемые (особенно взрослые) предпочитают отвечать на вопросы машине, а не живому человеку. В связи с этим вспоминается одна забавная ситуация, происшедшая с одним из авторов обучающей системы АБАК, когда он учился в институте. В конце семестра преподаватель, обращаясь к группе, спро-

сил, где они хотят сдавать семестровый зачет. Был выбор: либо сдавать как обычно преподавателю, либо в специализированном кабинете с техническими приспособлениями для опроса студентов. Группа дружно и эмоционально выбрала автоматизированный опрос. Преподаватель заинтересовался: «А почему вы предпочитаете автоматизированный опрос, ведь преподавателю зачет сдать гораздо проще?» Одна из студенток (кстати, отличница) на это ответила: «Преподавателю сдать, конечно, проще, но зато машине не стыдно смотреть в глаза».

Наконец, *в-девятых*, компьютер позволяет накапливать огромное количество разнообразных вопросов, что делает процесс обучения неповторимым и увлекательным.

Таким образом, сравнивая возможности учителя и компьютерной обучающей системы в плане задавания вопросов и ответной реакции на эти вопросы, мы приходим к выводу о неоспоримом преимуществе компьютера. Но самое примечательное то, что вопросы при компьютерном обучении превращаются из средства контроля знаний в основной дидактический прием обучения.

Рассмотрим следующую функцию средства обучения – «Показ обучаемому, как нужно делать». Понятно, что ни книга, ни компьютер ничего показать не могут, и здесь учитель находится на недостижимой высоте. Но показ необходим при обучении, скажем, игре на музыкальном инструменте, резьбе по дереву, шитью, вождению автомобиля и тому подобном, а при обучении работе с компьютером решающего значения не имеет.

Последняя функция обучаемого – это тренировка в предмете обучения. Здесь точно так же, как нет равных учителю в способности показывать, как нужно делать, так нет равных компьютеру в предоставлении возможности потренироваться

работать с собой. Остается здесь только вспомнить с нашей точки зрения совершенно справедливое мнение о том, что изучать работу на компьютере, не работая на нем, совершенно бесполезно.

Подводя итог ответа на вопрос «Как учить?», скажем, что обучающая система «АБАК» использует прежде всего то, что ее выгодно отличает от учителя и книги, а именно – развернутую систему разнообразных (по виду и содержанию) вопросов с мощной реакцией на неправильные ответы обучаемого, превращающие эти вопросы в средство глубокого усвоения учебного материала. Кроме того, поскольку некоторые виды вопросов достаточно затейливы по своей форме и способу ответов на них, то сама обучающая система как программный продукт несет в себе функции тренажера.

Кто будет делать наполнение?

На первый взгляд нет ничего проще, чем заполнить знаниями готовую «оболочку» обучающей системы. Надо пригласить преподавателей – одного, двух, или больше, на словах объяснить им тему учебного курса, и за короткое время они напишут все, что требуется. Но такое мнение оказалось абсолютно неверным. Практика показала, что далеко не каждый преподаватель (к сожалению) способен гладко, стройно и логично излагать свои мысли. Тот стиль чтения лекций, который «проходит» у доски в школьном классе или институтской аудитории, непригоден ни для книги, ни для компьютерной обучающей системы.

Основная трудность при создании обучающего курса состоит в том, что компьютер «запоминает» не только удачные преподавательские находки – аналогии, последовательность изложения материала – но также и все его ошибки. Те самые ошибки, на которые студенты просто не

обратят внимания или, в худшем для лектора случае, развеселятся. Оговорка преподавателя не выйдет из стен аудитории. Смысловую ошибку, например в выводе формулы, преподаватель найдет и исправит сам – на этой же лекции или на следующей – не важно. Компьютерная обучающая система, которая тиражируется в сотнях и тысячах экземпляров, которая распространяется (не бесплатно!) среди огромного количества пользователей, просто вынуждена претендовать на роль «абсолютной истины». Покупатели обучающих курсов совершенно справедливо требуют за свои деньги достаточно хорошего обучения. А чтобы написать и многократно проверить большой объем материала, нужно затратить изрядные усилия.

Еще один «камень преткновения» – это вопросы и реакции на ответы обучаемого. Придумать вопрос – большой труд, не зря задачки составляют, как правило, авторские коллективы. Но вдесятеро более сложная задача – на каждый из возможных ответов ученика написать так называемую «реакцию» компьютера, объясняющую характер ошибки в ответе и дающую именно тот дополнительный материал, который был недостаточно понят обучаемым.

И, наконец, тот человек, который пишет «наполнение», кроме блестящего владения предметом обучения, должен, естественно, владеть и основами программирования, так как обучающий курс является некоторой программой на специальном языке.

Как далеко выше сказанное от прежнего суждения, что де обычный преподаватель, оставаясь на час-два после занятий, может создать обучающий курс! Практический опыт показывает, что только команда профессионалов, оснащенная

всеми необходимыми техническими и программными средствами, способна создать хорошее «наполнение» обучающей системы.

Заключение

Система АБАК является сложной программой, которая управляет огромными объемами обучающей информации. Это добрый, но настойчивый экзаменатор, который будет раз за разом задавать вам вопросы и подробно объяснять, где вы допустили ошибку. Система АБАК помнит все изученные вами темы, все вопросы, на которые вы дали правильные ответы, и не станет задавать их снова. Важнейшее достоинство АБАК – развитая подсистема задания вопросов и анализа ответов.

Программа задает вопросы всех видов, которые только можно придумать, и реагирует на каждый вариант ответа обучаемого особым образом. Все тексты, все вопросы, все реакции – это более 30000 строк текста, введенного программистами объединения «АБАК», многократно просмотренного и отлаженного. В обучающую систему АБАК вложено около двенадцати человеко-лет упорного труда, и в результате мы имеем лучшую в России обучающую программу, которая, наконец, действительно способна учить!

Обучающая система АБАК, несомненно, явится незаменимым помощником для всех тех, кто приступает к овладению новыми информационными технологиями, будь то руководитель предприятия, работник бухгалтерии, студент учебного заведения, работник малого коммерческого предприятия – словом, для всех тех, кто осваивает персональный компьютер и желает этот процесс сделать приятным, но, что самое главное, максимально быстрым и эффективным.

ВСЕМ РУКОВОДИТЕЛЯМ ШКОЛ, ПТУ, ТЕХНИКУМОВ, ЛИЦЕЕВ

*Избавьте себя от тяжелой, нервной и изматывающей работы
по составлению расписаний учебных занятий.*

Работа диспетчера недостойна Вас

Передайте эту работу компьютеру

НОВЕЙШАЯ СИСТЕМА СОСТАВЛЕНИЯ РАСПИСАНИЙ

Альтернатива • Расписание

РЕШИТ ВАШИ ПРОБЛЕМЫ

Она позволит Вам:

- автоматически строить готовое расписание за минимальное время (порядка 15 минут);
- гибко учитывать все требования к расписанию (гигиенические, методические и т.д.);
- учитывать пожелания учителей и обеспечивать отсутствие «окон»;
- планировать занятия любого типа (поточные лекции, с разделением на подгруппы и т.д.);
- быстро сделать временную коррекцию расписания в случае особых обстоятельств (болезни учителей, срыва занятий и т.д.);
- осуществлять автоматическое слежение за выполнением учебного плана.

Система работает на любом IBM-совместимом компьютере.

Ведутся работы по переносу программы на УКНЦ, ДВК-3 и «Корвет».

НЕ ТЕРЯЙТЕ ВРЕМЕНИ!

ВЫШЛИТЕ НАМ ЗАЯВКУ СЕГОДНЯ ЖЕ!

НАШИ ЦЕНЫ САМЫЕ НИЗКИЕ (~ 50 тыс.руб.)

ФИРМА-РАЗРАБОТЧИК — НТП «НОВАЯ АЛЬТЕРНАТИВА»

Адрес: 656057, г.Барнаул, а/я 2513.

Директор Гриценко А.Н.

ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР

К. А. Шульман

Координатор проекта «Образование» фирмы IBM

КОМПЬЮТЕРЫ IBM PS/1 — В РОССИИ

Монпелье во Франции, Санта-Паломба в Италии, Гринок в Шотландии... Названия этих городов хорошо известны пользователям вычислительной техники в Европе: в них расположены заводы корпорации IBM по производству компьютеров различных типов. Недавно этот ряд пополнил подмосковный Зеленоград — в октябре 1993 года на заводе «Квант», входящем в объединение «Научный Центр», была открыта линия по производству персональных систем IBM PS/1. Произошло событие, имеющее принципиальное значение как для «Кванта», так и для IBM: завод начал выпускать персональные компьютеры, пользующиеся колоссальным спросом в Европе, а IBM первой из известных компьютерных фирм стала не только поставлять, но и производить персональные компьютеры в России.

В настоящее время корпорация IBM выпускает около 20-ти различных типов компьютеров IBM PS/1. В первое время завод «Квант» будет выпускать одну-две модификации младших моделей серии IBM PS/1. Эти компьютеры практически не будут отличаться от аналогичных, выпускаемых в Гриноке. Одна элементная база, неизменно высокие стандарты качества, специальная подготовка пер-

сонала «Кванта» — именно благодаря этому зеленоградские компьютеры будут выпускаться с логотипом IBM и иметь стандартные гарантии фирмы (1 год — минимум, до 3-х лет — по желанию заказчика). Разница будет лишь в программном обеспечении: все программные пакеты и документация будут русифицированы. Что же касается цены, то стоимость IBM PS/1 из Зеленограда на российском рынке должна быть существенно ниже, чем стоимость аналогичных моделей из Шотландии. Распространение и техническое обслуживание этих компьютеров будет производиться, в основном, через сеть бизнес-партнеров фирмы IBM.

Персональные компьютеры серии IBM PS/1 (Personal System/1) появились на европейском рынке примерно два года назад, при этом они были разработаны как компьютеры для малого бизнеса, образования и домашнего использования. Эти компьютеры удачно сочетают традиционно высокое качество IBM и доступную цену. Компьютер прост в установке, а относительно небольшие размеры системного блока (примерно 15 кв. дм) и клавиатуры позволяют сэкономить место на рабочем столе, что особенно важно для школ — ведь зачастую за одним столом работает двое учеников.

Персональные системы IBM PS/1 выпускаются в 2-х настольных и настольной модификации в зависимости от требуемого количества свободных 16-ти битных слотов расширения (от 3-х до 8-ми) и гнезд для дополнительных устройств ввода-вывода (от 1-го до 4-х). На системной плате устанавливается от 4-х Мб оперативной памяти и процессор Intel 486SX/25 и выше.

В стандартный комплект также входят:

- 14" SVGA монитор с поворотной подставкой;
- дисковод 3.5" 1.44 Мб;
- жесткий диск ;
- русифицированная клавиатура и мышь;
- комплект программного обеспечения и документации.

Возможна также установка дополнительного дисковода 5.25" 1.2 Мб, дополнительных винчестеров, стримера. Кроме того, выпускается вариант Multimedia: стандартный компьютер IBM PS/1 дооснащается аудиоплатой, устройством воспроизведения компакт-дисков, выносными колонками с регулятором громкости и микрофоном, а комплект программных средств дополняется пакетом игровых и учебных программ на компакт-дисках.

В процессе производства IBM PS/1 комплектуются программными пакетами: операционной системой IBM DOS 6.0, операционной оболочкой Microsoft Windows 3.1, интегрированным пакетом Microsoft WORKS for

Windows 2.0 (база данных, текстовый редактор, электронные таблицы), а также антивирусными и диагностическими программами. Все эти программные пакеты снабжены исчерпывающей документацией и заранее установлены на винчестере. Это позволяет начать работу сразу же после подключения компьютера к электрической сети, избежать затрат времени и средств на закупку и установку программного обеспечения.

У читателя может возникнуть вопрос: не слишком ли мощный компьютер фирма IBM предлагает для системы образования? Все зависит от того, как компьютер будет использоваться на уроке. Конечно, применение IBM PS/1 для изучения темы «Удаление и копирование файлов в системе MS DOS» равносильно стрельбе из пушки по воробьям. Следует, однако, учитывать современные тенденции в развитии учебного программного обеспечения. Все более активное применение звука, живого и неподвижного видеоизображения в учебных программах требует наличия средств для их обработки и хранения (мощных процессоров, винчестеров и/или проигрывателей компакт-дисков) на каждом рабочем месте.

Более подробную информацию о персональных системах IBM PS/1 можно получить в Московском представительстве фирмы IBM:

113054 Москва, ул. Бахрушина, 18.
тел. (095)235-66-12, факс 235-48-49.

А. В. Зимарев, А. А. Данилин

фирма «F-Bit»

СИСТЕМА ПОДГОТОВКИ КОНТРОЛИРУЮЩИХ ПРОГРАММ AUTO CONTROL

Одним из основных типов программного обеспечения образования являются системы контроля знаний. Если обучение с помощью компьютера вызывает у некоторых преподавателей определенный скептицизм, то реальную возможность компьютеризации контроля вряд ли кто будет отрицать. Для этого существует много программных средств, однако, сколько программ – столько достоинств и недостатков. Что же выбрать?

Попробуем обрисовать контуры системы контроля знаний, которая удовлетворяла бы большинству необходимых требований.

Во-первых, система должна обеспечивать контроль по любой дисциплине. Казалось бы, такого не может быть, ведь нельзя сделать программу, которая проводила бы контроль и по русскому языку, и по математике. Поэтому и существует такое многообразие узко специализированных систем контроля. Но оказывается ничего невозможного здесь нет, и вместо покупки нескольких программ можно обойтись лишь одной. Что же это за системы? Универсальностью обладает лишь один класс систем контроля – оболочки для создания тестов.

Следующее требование. Если уж мы остановились на конструкторах тестов, то такая система должна предоставлять широкие возможности для создания вопросов по различным предметам, которые могут сильно отличаться как по своей структуре, так и по сложности. Иными словами, система должна иметь возможность создавать вопросы различных типов.

Далее третье требование. Не следует упускать из виду фактор секретности и защиты от «ложных знаний». То есть в системе обязательно должны присутствовать такие элементы, как ограничение до-

ступа к уже готовым тестам и динамическое изменение вопросов в процессе контроля так, чтобы для разных учеников один и тот же вопрос выглядел по-разному. Кроме того, для выполнения третьего требования необходимо наличие в системе нескольких способов опроса, в том числе со случайной выборкой вопросов из теста.

Судя по приведенным выше минимальным требованиям, которым должна удовлетворять хорошая система контроля знаний, обеспечит такой уровень не просто. И кажется, что такая система будет сложной, большой, и дорогой. Но нет, теперь мы можем сказать, что есть система для создания тестов, которая отличается широким набором возможностей в сочетании с отличным интерфейсом, небольшим объемом, подробно составленным руководством и сравнительно невысокой ценой. Представляем эту систему: Auto Control 2.0, продукция фирмы F-Bit, занимающейся разработкой программного обеспечения для образования.

Auto Control предоставляет пользователю широкие возможности в создании разнообразных тестов. В системе предусмотрены три типа вопросов – с выбираемым ответом, со свободно конструируемым ответом и вопрос-задача с генерируемыми данными. При этом вопросы могут быть как текстовыми, так и графическими. Рассмотрим все типы вопросов подробнее.

Вопрос с выбираемым ответом достаточно традиционен для систем контроля знаний и вообще широко применяется в тестах. Выглядит он следующим образом: предлагается вопрос и несколько вариантов ответов, среди которых один или несколько верны. Задача состоит в том, чтобы ввести номера верных вариантов. Auto Control позволяет определять, ва-

жен ли порядок вариантов в ответе, а также производит перемешивание вариантов при контроле – один из способов защиты от «ложных знаний».

Вопрос с конструируемым ответом тоже довольно распространен. Предлагается только вопрос, а ответ вводится контролируемым полностью в виде слова или нескольких слов. При этом имеется возможность задавать синонимы к словам эталонного ответа, а также определять важность порядка вводимых слов.

И, наконец, третий тип вопроса – задача. Такой вопрос очень похож на предыдущий тип за исключением малого – в текст вопроса можно вставить переменные, которые при контроле будут заменены на конкретные значения.

Составитель вопроса сам определяет тип переменной и диапазон генерации значений. Соответственно эталонным ответом на задачу будет не число, а формула для расчета верного ответа. Такой воп-

рос, как и вопрос с выбираемым ответом, будет в различных случаях выглядеть по-разному и иметь разные (с точки зрения контролируемого) ответы.

Кроме гибкости в создании вопросов, Auto Control предоставляет возможность настраивать параметры самого теста – способ выбора вопросов (случайно или по порядку), количество задаваемых вопросов (все или только часть), способ оценки ответов (четырёхбалльная, зачетная и рейтинговая системы) и другие параметры.

И последнее. Auto Control, в отличие от многих подобных разработок, имеет удобный интерфейс, отвечающий требованиям IBM Standard User Interface, мощный и удобный графический редактор для создания вопросов в графике и подробное руководство пользователя, которое делает освоение системы легким и доступным каждому.

103498 Москва, К-498, а/я164 «F-BIT»
тел.535-2222, 534-4831, факс 534-4832



Официальный ресейлер
корпорации **Borland**
в области образования

Лицензионные поставки:

- Borland Pascal 7.0 (рус.версия)
- Borland C++ 4.0
- Borland Paradox 4.0
- Borland Paradox для Windows
- Quattro Pro 5.0 (рус.версия)

Специальная цена для организаций образования !



535-2222, 534-4831, 534-4832 (факс)

АРХИВИРУЮЩИЕ ПРОГРАММЫ И РАБОТА С НИМИ

От редакции

НАЗНАЧЕНИЕ АРХИВАТОРОВ

При работе на IBM PC/XT/AT и совместимых с ними ПЭВМ часто возникает необходимость долговременного хранения тех или иных файлов, например, резервных копий наиболее важных программ. Как правило, библиотека резервных копий создается с целью обезопасить себя от потери особо ценных программ и данных, обычно хранящихся на жестком диске, при его поломках, из-за заражения компьютерными вирусами, из-за ошибок пользователя или по какой-либо другой причине.

Если количество файлов, подлежащих резервированию, невелико, для создания библиотеки резервных копий используются, как правило, команды-утилиты COPY и DISKCOPY (для резервирования информации, хранящейся на гибком диске). Однако при этом резервные копии занимают столько же места, сколько и их оригиналы. Тогда для резервирования достаточно большого количества файлов потребуется очень много дискет. Например, резервирование всей информации, записанной на жестком диске емкостью 20 Мб, потребует, как это легко подсчитать, 60 дискет по 360 Кб. При таком количестве дискет велика трудоемкость создания и периодического обновления «библиотеки».

Для таких случаев созданы специальные программы-архиваторы. Эти программы при работе производят сжатие данных и соответственно файлов (что дает в среднем экономию места около 40 – 50%). Кроме того, при сжатии (архивировании) файлы можно объединять в один блок – архивный файл, что обычно делается при «закладке на хранение» групп совместно используемых файлов. После этого с полученными архивными файлами можно обращаться, как и с любыми другими, используя средства DOS или, например, Norton Commander: копировать, переносить из каталога в каталог (MOVE), удалять, переименовывать и т.п. При необходимости требуемый архивный файл «распаковывается» с помощью программы-разархиватора (часто функции архивирования и разархивирования совмещены в одной программе), после чего архив превращается в тот же исходный файл или группу файлов, которые были в него помещены при архивировании.

Архивный файл содержит, кроме заархивированных файлов, а также их список (оглавление архива) и контрольные коды каждого файла для проверки целостности архива.

ВИДЫ АРХИВИРУЮЩИХ ПРОГРАММ

Из существующих сегодня программ-архиваторов наиболее широко распространены следующие: программа LHARC (автор Аруясу Ёсизаки, Япония) и ее версия LHA, пакет программ PKZIP (архиватор PKZIP и разархиватор PKUNZIP, PKWARE, Inc., США), программа-архиватор ICE.

В дополнение к пакету PKZIP имеются также программы PKZIPFIX (восстановление архивных файлов, содержащих ошибки) и ZIP2EXE (создание самораспаковывающихся архивных файлов).

Принципы использования большинства архиваторов сходны между собой. Различия сводятся, в основном, к формату записи команд и к формату выводимой на экран справочной информации.

ПОНЯТИЕ «БАЗОВОГО» КАТАЛОГА

«Базовым» будем называть каталог, в котором содержатся файлы, подлежащие архивированию, или в который файлы должны быть помещены после распаковки архива. Как правило, архиватор вызывают непосредственно из архивируемого каталога, и тогда понятия базового и текущего каталога совпадают. В отдельных случаях базовый каталог может быть указан в записи команды на архивирование/разархивирование с помощью пути доступа к нему (path).

Все файлы, содержащиеся в базовом каталоге (а также, если это необходимо, и если дан соответствующий формат команды, и в его подкаталогах), при архивировании преобразуются в один архивный файл или могут быть добавлены к уже существующему архивному файлу. При желании можно указать сразу несколько базовых каталогов, тогда в один архивный файл будут собраны все файлы из этих каталогов. При разархивировании архивного файла все содержащиеся в нем файлы помещаются в базовый каталог (если необходимо, в нем при этом могут быть автоматически созданы требуемые подкаталоги). Можно задать и другой режим работы разархиватора, тогда распакованные файлы будут «раскиданы» по нескольким базовым каталогам.

Примечание. «По умолчанию» в архив заносятся все файлы из указанного каталога. Если же нужно внести в архив только часть из них, используется явное указание имени файла (или списка имен). Кроме того, разрешается использовать шаблоны DOS (знаки «*» и «?») для указания группы файлов, например:

*.EXE — все файлы данного каталога с расширением EXE;

LESSON???.TXT — все тексты уроков (LESSON), независимо от их порядкового номера (трехзначного);

K1533LA3.* — все файлы (описания, модели и т. п.), касающиеся схемотехнического элемента K1533LA3 (например, в PCADe);

. — все файлы из данного каталога (как и по умолчанию).

Для выполнения любой из операций по архивированию, разархивированию, просмотру, контролю архивного файла и т. д. необходимо ввести имя программы-архиватора или разархиватора, команду (символ, указывающий архиватору, какая из операций требуется в данном случае), ключ (называемый также «расширением» или «опцией»), имя архивного файла и пути доступа к нему. Ключи и список архивируемых файлов являются необязательными параметрами и используются не всегда. В некоторых случаях может быть не указана и команда, тогда операция считается заданной по умолчанию.

Все архиваторы содержат в себе текст-подсказку для пользователя, которая может быть выведена на экран при задании определенных ключей или при вызове программы-архиватора без параметров (по одному только ее имени), то есть, по умолчанию.

Так, архиваторы ICE и LHARC выдают на экран подсказку при вызове их без параметров, а в архиваторе PKZIP и разархиваторе PKUWZIP для получения подсказок нужно ввести PKZIP/h или PKVNZIP/h соответственно, о чем указано в сопровождающем тексте при архивировании и разархивировании.

В подсказке указывается общий формат записи строки и список команд и ключей, доступных в данной программе. Например, для LHARC общий формат записи строки следующий:

LHARC_<команда> /[- , +, 2, <ключ>]_<имя архива>_<базовый каталог><файлы>.

В одной статье трудно описать все возможности, имеющиеся в архиваторах LHARC, KHA, PKZIP. Поэтому здесь расскажем об архиваторе LHA — описания его последних версий в отечественной литературе не существует. Желаящим глубже изучить методы применения программ-архиваторов PKZIP и PKUNZIP можно посоветовать обратиться к литературе:

Фигурнов В.Э. IBM PC для пользователя. М.: Финансы и статистика, «Компьютер Пресс», 1991 г.

Е.И. Агеенко

АРХИВАТОР LHA (Версия 2.11)

ВВЕДЕНИЕ

Задача архивирования (сжатия) программ и файлов данных возникла с появлением ЭВМ. Но только с появлением персональных компьютеров с их ограниченной памятью и резким ростом числа программ она встала наиболее остро. Первые архиваторы работали под управлением CP/M, и их архивы назывались библиотеками. С появлением MS-DOS собственно и появилось понятие «archive – архив», так стали называться файлы, содержащие сжатую информацию. Первые архивы имели расширение «.ARC», которое с появлением различных архиваторов стало видоизменяться. Так появились: .PAK, .ZIP, .ARJ, .ICE, .LZH архивы. В настоящее время наиболее популярны архиваторы: ZIP (PkZip, PkUnZip), ARJ, LZH (Lha). Первый быстро работает (и неофициально является американским стандартом сжатия информации), второй делает наиболее плотно сжатые архивы. Его отличительная особенность – возможность резать длинные архивы на части. Но наиболее удобным является LHA, имеющий приемлемую скорость работы и чуть уступающую ARJ плотность сжатия. В Японии архиватор LHA является стандартом, в других странах стал общеизвестным и часто используемым.

Последняя известная автору версия – это 2.13, но мы рассмотрим аналогичную ей 2.11 (от 3 мая 1991 г.). LHA 2.11 – это пересмотренная версия более медленного архиватора LH113c.EXE. Так как MS-DOS 5.0 теперь имеет внутреннюю команду LH (для загрузки в High Area), имя изменилось на LHA.EXE. Над этой новой версией работали в течение двух лет, начиная с момента выпуска LH113c.EXE. Архиватор LHA 2.11 мы далее и будем рассматривать.

СОБСТВЕННО LHA

Отличия

LHA лучше, чем LH версии 1.13 по коэффициенту сжатия, особенно для боль-

ших файлов. Имеется небольшое количество исключений, если использовать маленькие файлы (до 1 кб). LHA более быстрый, чем LH113c в разархивирующем режиме, так как применяет алгоритм статического кодирования Хоффмана. По сравнению с более старым динамическим кодом Хоффмана в LHA, последний нуждается в большем количестве памяти по сравнению с LH113c. Если не имеется достаточного пространства для работы LHA, он все равно продолжит выполнение, хотя скорость сжатия может уменьшиться. LHA совместим с предыдущими версиями снизу вверх. Используйте опцию «/o», и вы сможете делать архивы, совместимые с версией LHarc 1.xx – иначе LHarc будет жаловаться на неизвестный метод.

LHA распространяется как свободная программа с зарезервированным авторским правом.

Общий формат :

LHA <command> [/option[+012|WDIR]] <archive[.LZH]> [DIR\] [filenames]

- Команда <command>: если не используется, вы получаете справку по работе с программой.
- Опция <option>: вы можете использовать одну или более опций поясненных ниже (символ «/» или «-» обозначает начало опции).
- Архив <archive>: имя архива (составное имя). По умолчанию принимает расширение «.LZH». Если вы определите другое расширение, LHA будет запрашивать у вас подтверждение его использования: «Расширение архива не ".LZH." Продолжить обработку? [Y/N]».
- Групповые символы (такие, как ? и *) допустимы в имени архива с командами «e», «x», «p», «l», «v» и «s»: «LHA e *.LZH *.c» извлекает все файлы с расширением .c из всех архивов в текущем каталоге.
- Каталог <DIR>: имя базового каталога.
- Имена файлов <filenames>: список имен файлов (составных). Если имена файлов не определены, LHA будет обрабатывать все файлы без имен каталогов. Групповые символы «*» и «?» используются аналогично MS-DOS. Вы не можете использо-

вать их, чтобы определять каталоги, за исключением переключателя /r2.

- WDIR : имя рабочего каталога.

Команды (<command>)

A (Add – Добавление) – сжатие и добавление файлов к архиву

LHA архивирует и добавляет определяемые файлы к архиву. Если именованный архив не существует, тогда создается архив с этим именем. LHA перезаписывает любой файл с данным именем, уже существующий в архиве (сравните с командой «U»).

Команды «A» и «M» используются для создания архива.

Пример 1: LHA a EX *.EXE – LHA делает архив с именем EX.LZH из всех файлов с расширением «EXE». Расширение .LZH устанавливается по умолчанию.

U (Update – Модификация) – сжатие и модификация

LHA сжимает и добавляет определяемые файлы к архиву (как команда «A»). Но если LHA обнаруживает в архиве файл с тем же именем, то из них выбирается более поздний по времени создания, чтобы добавить к архиву.

Пример 2: LHA u EX *.C – все файлы с расширением «.C» архивируются в EX.LZH. Если вы уже имеете архив EX.LZH, содержащий файлы с совпадающими именами, то из них выбираются более новые.

M (Move – Перемещение)

Эта команда работает аналогично команде «u». Отличие заключается в том, что после архивирования файлы удаляются. (LHA не может удалять каталоги – это вы сделаете сами.)

Пример 3: LHA m EX *.C – делает то же самое, что и две команды «LHA u EX *.C», «Del *.C».

Если использована опция «/C», то LHA перемещает весь файл в архив, игнорируя временные метки. Вы можете потерять файл со старой временной меткой.

F (Freshen – Обновление)

LHA ищет для обновления файл с тем же самым именем, какое он имеет в архиве. Если найден более поздний файл, то LHA перезаписывает его в архив. Но при

вызове «LHA f /c ARC» LHA не будет проверять временную метку, а перезапишет все файлы архива.

Пример 4: LHA f EX LHA.DOC – пусть файл «LHA.DOC» сжат и заархивирован в архиве «EX.LZH». Если у вас имеется более новый файл «LHA.DOC», тогда он записывается в архив на место старого.

D (Delete – Удаление)

LHA удаляет файл (или файлы) из архива.

E (Extract – Извлечение)

LHA извлекает файлы из архива. Если в текущем каталоге вы имеете файл с таким же именем, но более поздним временем создания, то LHA не будет разархивировать этот файл.

LHA извлекает файлы из архивов, сделанных L113C только с идентификаторами сжатия lz3, lz4, lz5.

Пример 5: LHA e EX LHA – извлекает все файлы из архива «EX.LZH».

Пример 6: LHA e EX *.COM – LHA извлекает все файлы с расширением .COM из «EX.LZH».

X (eXtract with pathnames – Извлечение с подкаталогами)

LHA извлекает сжатый файл с маршрутом к нему. Ненайденные каталоги создаются заново. Если была использована архивация с полными именами путей (/x1), LHA восстанавливает всю структуру каталогов.

Пример 7: LHA e /x1m1 EX – делает то же самое, что и «LHA x EX». Предположим, что «\BIN\CG86.EXE» был среди файлов архива «EX.LZH». Тогда LHA в случае необходимости создаст каталог «\BIN» и извлечет «CG86.EXE» туда.

P (disPlay – Печать файлов)

Печать на внешнее устройство. LHA посылает разархивированный файл из архива на стандартное устройство вывода.

Пример 8: LHA p ex lha.doc – LHA извлекает «LHA.DOC» из «EX.LZH» и посылает его на ваш терминал.

Пример 9: LHA p EX LHA.DOC > prn – LHA извлекает «LHA.DOC» из «EX.LZH» и посылает его на ваш принтер.

L (List of files – Список)

LHA печатает имена файлов из архива. В распечатке файл, архивированный с маршрутом, будет иметь метку «+» в начале строки. Если использовать вызов «LHA L /x ARC», то вы получите полные маршруты к файлам архива. (Задание команды L для получения списка необязательно.)

Пример 10а: LHA LHA210.LZH – LHA принимает команду «L» по умолчанию, в результате будет показан список архивированных файлов.

Пример 10б: LHA /x LHA210.LZH – вы получите подобный список с полными маршрутами к файлам. Каждый файл при этом отобразится на двух строках.

ВНИМАНИЕ: косая черта «/» используется как разделитель имен маршрутов.

V (View listing – Просмотр)

«LHA v ARC» – то же самое, что и «LHA l /x ARC».

T (Test – Тест)

Проверяет целостность архива путем проверки контрольных сумм (CRC).

Пример 11: LHA t LHA.EXE или LHA t LHA206.EXE – будет проверяться подлинность программы, которую вы имеете. Вы не можете проверить LHA.EXE, если использованы компрессоры исполняемого файла типа LZEXE, PKLITE или DIET.

Пример 12: LHA t EX LHA – проверяется целостность LHA в «EX.LZH».

S (make SFX – Самораспаковывающийся (SFX) архив)

LHA делает архив самораспаковывающимся. По умолчанию принимается переключатель /x0.

SFX, сделанный с переключателем /x0 (маленькая модель), извлекает файлы в текущий каталог. Размер SFX-файла с маленькой моделью меньше, чем сделанного с переключателем /x1 (большой модели). Если вы хотите использовать некоторые возможности SFX-архива, используйте переключатель /x1. Большая модель SFX также имеет способность восстанавливать структуры с подкаталогами.

Пример 13: LHA s EX – LHA делает маленькую модель EX.EXE из EX.LZH.

Переключатели (</option>)

Каждый переключатель имеет три числовых значения: 0, 1 или 2 для определения вида действия. Для некоторых опций значения 1 и 2 идентичны.

Вы можете переключать 0 или 1 как «+» или «-» для LH113с. Вы также можете изменить символ, предшествующий опции, с «/» на «-».

/x[0|1] (eXtended – расширенный)

С этой опцией LHA использует расширенные имена файла, а именно полные маршруты для файла. По умолчанию LHA сохраняет только имена архивируемых файлов и игнорирует имена каталогов, в которых они находятся.

Пример 14: предположим, что вы находитесь в корневом каталоге, и у вас есть два файла «stat.h» и «\tc\include\sys\stat.h». Команда: «LHA ARC.LZH stat.h» добавляет только файл «stat.h» из текущего (корневого) каталога к «ARC.LZH», но команда «LHA ARC.LZH stat.h \tc\include\sys\stat.h» заставит LHA прервать работу с сообщением «Те же самые имена с другим маршрутом», потому что имена каталогов по умолчанию игнорируются при архивировании файлов. Однако «LHA /x ARC.LZH stat.h \tc\include\sys\stat.h» добавит оба файла к «ARC.LZH» со своими маршрутами. Созданный таким образом архив раскрывается командой «x».

/p[0|1|2] (full Pathnames – Точность)

Задаёт точный поиск для имен файлов. Предположим, что архив «TC.LZH» содержит и «STAT.H», и «SYS\STAT.H». Простая команда «LHA e TC stat.h» будет извлекать оба файла в текущий каталог. Чтобы избежать такого беспорядка, вы можете напечатать «LHA E -P TC stat.h», чтобы извлекать только «STAT.H». В то время как, печатая: «LHA E -P TC sys\stat.h», получите «SYS\STAT.H».

/c[0|1|2] (skip time Check – Игнорирование сравнения по времени)

С командами «u», «f», «e», «x» LHA проверяет временные метки файлов. По умолчанию LHA выбирает самый новый файл. Эта опция позволяет LHA игнорировать временные метки.

/m[0|1|2] (no Message)

Команда «LHA e /m2 ARC.LZH ФАЙЛЫ» разархивирует все файлы, при этом каждый повторяющийся файл получит новое, еще не использованное расширение между 000-999.

/a[0|1] (any Attributes – Любые атрибуты)

Этот переключатель дает возможность LHA архивировать файлы с любыми атрибутами. В процессе архивирования с переключателем /a0 (по умолчанию) LHA не будет архивировать файлы с атрибутами Hidden (скрытый) и System (системный). Файл с атрибутом «только для чтения» архивируется с этим же атрибутом. В процессе разархивирования с /a1 разархивированные файлы сохраняют первоначальные атрибуты. С /a0 вы не можете разархивировать файлы со скрытым и системным атрибутами. Файлы «только для чтения» будут разархивированы лишеными их первоначального атрибута.

/r[0|1|2] (Recursively – Рекурсивность)

Этот переключатель дает возможность LHA извлекать архивируемые файлы рекурсивно из подкаталогов. «Рекурсивно» – значит, что LHA ищет все файлы из всех подкаталогов в данном каталоге, если таковые имеются. Существуют три различных режима для переключателя «/r»:

- /r0 – (нерекурсивный режим, по умолчанию) LHA собирает файлы, определяемые только составными именами.
- /r1 – LHA разделяет данный маршрут на имя каталога и имена файлов. LHA рекурсивно собирает файлы с данными именами из всех каталогов.

Пример 15: LHA A /r1 SOURCE.LZH \SOURCE *.C \SOURCE\ *.H LHA – собирает файлы с расширением C и H из каталога «\SOURCE» и подкаталогов, вероятно «\SOURCE\SAMPLES\ *.C», но не «\SOURCE\ *.OBJ».

- /r2 – LHA рекурсивно собирает все файлы из всех определяемых подкаталогов. Структура подкаталогов определяемого каталога архивируется как она есть.

Пример 16: LHA a EX /r2x1 a: \ *. *

ВНИМАНИЕ: LH113с устанавливает «/x» всякий раз, когда «/r» устанавливается в «e» или «x» команде, в отличие от LHA.

Пример 17: LHA a /r2x1 LZH a:\ – архивируется весь диск A с сохранением структуры подкаталогов.

/w[0|1]<Directory Name> (Work directory – Рабочий каталог)

Определяет имя каталога, где LHA делает временные файлы:

«LHA a /wd:\ ARC.LZH ФАЙЛ».

По умолчанию LHA делает все временные файлы в каталоге, где должен находиться ARC.LZH. LHA делает временные файлы на текущем каталоге, если рабочий каталог никаким именем не определяется, например «/w1». Когда вы устанавливаете относящуюся к окружению переменную «TMP», этот переключатель автоматически устанавливается в 1. Переключатель помогает, когда вы не имеете достаточного количества памяти в вашем базовом каталоге, или когда вы имеете высокоскоростное устройство памяти, подобное псевдодиску.

/n[0|1|2] (No indicator – Без индикации)

В этой версии LHA в процессе сжатия выводит на экран индикатор «oo.....». Переключатель должен подавлять этот вывод.

- /n1 – LHA отключает вывод «oo.....», чтобы указывать прогресс.
- /n2 – LHA отключает вывод имен файлов и процента сжатия.

/t[0|1] (Time stamp – Временная метка)

С этим переключателем и командами «a», «u», «m», «f», «d» временная метка архива устанавливается согласно самому новому файлу в архиве. По умолчанию («/t0») временная метка архива – время, когда он создается.

/z[0|1|2] (Zero compression – Без сжатия)

С этим переключателем LHA делает архив без сжатия:

- /z1 – Ни один файл не сжимается.
- /z2 – Архивируется все, кроме файлов с расширениями .ARC, .LZH, .LZS, .PAK, .ZIP, .ZOO.

- /z <расширение> – не сжимаются файлы с определяемым расширением. Вы можете использовать групповой символ, чтобы определить расширение, и помещать «.» перед расширением. Так, с «/zdbf» вы не сжимаете файлы с расширением «.DBF». Вы можете определять расширения многократно, записывая последовательно:

«LHA A /ZEXE /ZCOM ARCHive.LZH *.*».

В этом случае в текущем каталоге архивируются и упаковываются все файлы, кроме имеющих расширение «COM» или «EXE».

/o[0]1 (Old compatible compression – Совместимое архивирование)

LHA делает архив совместимым с форматом LH113с. Даже в этом случае LHA делает лучшее сжатие, чем LH113с версии 1.хх. Идентификатор заголовка автоматически устанавливается в «/h1».

/h[0]1[2] (Header level – Уровень заголовка)

Выбор уровня заголовка, по умолчанию – /h0.

/i[0]1 (don't ignore case – Различать регистры)

LHA распознает верхний и нижний регистры. Эта опция предназначена для распаковки архивов, сделанных в другой OS, отличной от MS-DOS. В версии LHA для MS-DOS LHA не может отличить верхний и нижний регистры, когда архивирует файлы. Все имена сохраняются в верхнем регистре.

/l[0]1[2] (Long display – Подробный вывод).

LHA выводит имена файлов в различных форматах при архивации и разархивации:

- /l0 – только имена файлов.
- /l1 – сохраняемые полные маршруты.
- /l2 – сохраняемые полные маршруты с родительским каталогом.

Пример 18: LHA a /r1x112 LINK.LZH c:\LINK.*

/- [1] (Переключатель первого символа)

LHA распознает символы «-» и «@», как первые символы имени файла. По умол-

чанию любой файл, начинающийся с «@», подобно «@xxx» распознается как файл диалога xxx.

БАЗОВЫЙ КАТАЛОГ

Архивация

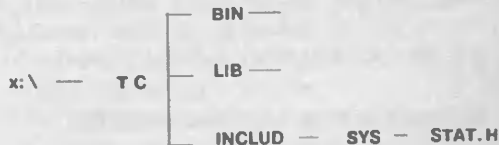
Если вы определяете имя базового каталога, тогда LHA будет относиться к нему, как будто это – текущий каталог. По умолчанию базовым является текущий каталог или, если вы находитесь на другом диске по отношению к архивируемым файлам, – корневой каталог. Определяйте окончание имени базового каталога символом «\». Все файлы будут архивироваться без сохранения имени базового каталога, даже если вы используете переключатель /x или /r.

Пример 19: Предположим, что вы имеете следующее дерево каталогов:

Выполнение «LHA a /r ARCHIVE.LZH tc\include\sys\stat.h» из корневого каталога приведет к добавлению файла «\tc\include\sys\stat.h» к «ARCHIVE.LZH» с маршрутом «sys\stat.h».

ВНИМАНИЕ: команда «LHA a /r ARCHIVE.LZH tc\include\» добавит к архиву все файлы из tc\include\ с маршрутами относительно его самого (как если бы он был текущим). Здесь «tc\include\» – базовый каталог. Но команда «LHA a /r ARCHIVE.LZH tc\include» добавит к архиву файлы с полными маршрутами (если вы находились в корневом каталоге). Здесь «tc\include» – составное имя файла, аналогичное «tc\include*.*».

Разархивация



Базовый каталог – это каталог, в который LHA извлекает файлы. По умолчанию является текущим каталогом. Вы можете определить ряд каталогов как ваши базовые каталоги.

Пример 20: LHA x program c:\BIN\
*.EXE *.COM c:\TEMP*.MAN *.DOC

Предположим, вы находитесь на каталоге d:\. Вы хотите извлекать файлы с расширениями .COM и .EXE в c:\BIN, и с расширениями .MAN и .DOC в каталог c:\TEMP. Это эквивалентно следующему набору командных строк :

```
D>C:
C>CD \BIN
C>LHA x D:\program *.COM *.EXE
C>cd \TEMP
C>LHA x D:\program *.MAN *.DIX
```

Конечно, вы не можете разархивировать одиночный файл в несколько каталогов. Каталог, определенный раньше, имеет высший приоритет.

ПЕРЕАДРЕСАЦИЯ DOS И ФАЙЛ ДИАЛОГА

LHA не может принимать слишком большое количество имен файлов в командной строке из-за ограничения MS-DOS на ее длину. Избежать этого неудобства можно, используя «каналы ввода» и «переназначение» DOS. Кроме того, LHA может использовать рабочий файл, называемый файлом диалога, в котором указаны имена файлов, которые нужно архивировать.

Файл диалога

Имя файла диалога отмечается символом «@». Он должен быть текстовым и содержать имена необходимых файлов. В файле диалога игнорируется код CR (возврат каретки) и используется пробел как разделитель. Вы можете записывать опции в файле диалога, но не можете вложенно вызвать другой файл диалога.

Каналы DOS и переназначение

Наряду с файлами диалога вы можете использовать «каналы» и «переназначение» DOS. Вы можете создать файл диалога способом переадресации.

Пример 21: LHA I /n1 PARTS > FILE LHA a NEWPARTS @FILE — создает список файлов архива «PARTS.LZH» в файле FILE и использует его, чтобы сделать новый архив «NEWPARTS.LZH». Это возможно благодаря файлу диалога, который пол-

учен после некоторого редактирования файла FILE.

Переменные, относящиеся к окружению DOS

LHA и LHARC (параметры по умолчанию)

Вы можете устанавливать любой из переключателей LHA по умолчанию в системной переменной «LHA» в файле autoexec.bat: «LHA = /we: /r2 /x1» устанавливает рабочий каталог E:\ и опции /r2 /x1. Они будут использоваться по умолчанию. Переменная LHA отменяет LHARC.

TMP (рабочий каталог)

Вы можете определить рабочий каталог LHA в системной переменной «TMP»: «TMP=E:», но переключатель /w будет отменять эту установку.

Коды завершения

LHA возвращает следующие коды завершения после выполнения, которые вы можете использовать в командном BAT-файле:

- 0 — Нормальное выполнение.
- 1 — Ошибка CRC, вероятно с командами «e», «x», «t».
- 2 — Фатальная ошибка. Завершить процесс без транзакций.
- 3 — Неудача записи временных файлов в архиве. Вы можете найти временный файл LHTEMP)2(.LZH на вашем рабочем каталоге, переименовать его с расширением LZH, и использовать как архив.

Рабочие файлы

LHTEMP)1(.LZH — старый переименованный архив.

LHTEMP)2(.LZH — рабочий файл нового архива.

SFX—САМОРАСПАКОВЫВАЮЩИЙСЯ АРХИВ

SFX.EXE — исполняемый EXE-файл с сохраненными в архиве файлами, которые автоматически извлекаются при его выполнении. Только LHA.EXE делает SFX-файлы из архива *.LZH; сделанный LHA.EXE. Вы не можете делать исполняемый SFX-файл из архива, сделанный

LH113с.1.хх. LHA делает две модели SFX файлов :

маленькая модель (переключатель /x0).

большая модель (/x1).

Использование : SFX.EXE [/x] [!/]
[/eDIR] [DIR]

- /x : не создавать новые каталоги.
- /! : выполнять BAT-файл автовыполнения.
- [/eDIR], [DIR] : определяет каталог, для извлечения.

КОММЕНТАРИЙ

Любой файл с именем «!», входящий в состав архива SFX.EXE, отображается при разархивации. Вывод заканчивается простым сообщением «[Y/N]». SFX переходит к выполнению, если получает «Y». По команде «N» осуществляется выход (аварийное прекращение работы). Символ «!» использовался, поскольку это – самый первый ASCII-символ. Файл, начинающийся с «!», будет всегда первым в архиве, если вы, конечно, предварительно сортировали каталог по имени. Это может использоваться для объявления, приглашения к разархивации и т. д.

СПЕЦИФИКАЦИЯ КАТАЛОГА

Это возможно только для большой модели. Вы можете задать имя каталога, в который будут извлекаться файлы из SFX.EXE.

Пример 22: SFX.EXE /xe c:\user – LHA извлекает файлы из SFX.EXE в C:\USER. Вы можете также напечатать: «LHA e /x0 SFX.EXE c:\user». Это абсолютно аналогично.

АВТОВЫПОЛНЕНИЕ

Это возможно только для большой модели. Архивируйте командный файл с именем «!.BAT», и включите его в SFX файл. Этот файл выполнится, если вы напечатаете: «SFX -!». Любой существующий файл «!.BAT» перезапишется, поэтому вы не можете активировать существующий «!.BAT», используя «-!», даже если он имеется в текущем каталоге. Команд-

ный файл выполняется, только если он заархивирован в SFX-файле.

МОДЕЛИ SFX-ФАЙЛОВ

Существуют две модели SFX-файлов.

Маленькая модель («LHA s SFX.LZH»)

Это – модель по умолчанию. LHA игнорирует структуры каталогов, даже если вы создаете архив с опцией «/x» или «/r[1|2]». SFX-файл содержит только имена файлов и поэтому извлекает все файлы в текущий каталог. Нет никакого ограничения размера SFX.EXE, как это было у LH113с.

Большая модель («LHA s /x1 SFX.LZH»)

Большая модель SFX.EXE имеет несколько больший размер по сравнению с маленькой и содержит полные имена маршрутов к включенным файлам, как и ваш архив .LZH, поэтому при разархивации будет создана полная структура каталогов, даже если вы не используете опцию «/x».

ГЛАВНЫЕ ОТЛИЧИЯ ОТ LH113С

LHA имеет ряд принципиальных отличий от LH113с:

- Имена файлов не сортируются при архивации, а добавляются в порядке, в котором они записаны в каталоге (если используются групповые символы). Если вы определяете порядок файлов, то он воспроизводится в архиве. Если вы добавляете новые файлы, они добавятся в конец архива. Имеются внешние сервисные программы, написанные другими авторами, которые вы можете использовать, чтобы сортировать файлы в вашем каталоге до добавления их к архиву .
- В LHA переключатель /r не обязательно активирует /x при разархивации. В этой версии LHA команда «X» эквивалентна «E /x1m1», а именно, LHA восстанавливает всю структуру подкаталогов.
- Если SFX файл содержит !.BAT файл, он активируется ключом «-!» после имени исполняемого SFX-файла.
- Временная метка LHA считается относительно 1970-01-01, 00:00:00 UTC. Вы должны быть внимательны, чтобы не устанавливать дату или время, которое является иллюзорным, подобно 00-00-00. Это вызывает переполнение счетчика.

Н.Е. Платонов

CD ROM ТЕХНОЛОГИИ

В настоящее время достаточно широкое распространение в мире получил способ хранения информации на оптических компакт-дисках (Compact Disk – сокращенно CD). В бытовой технике это прежде



всего аудио- и видеозаписи (соответственно, CD A и CD V). В вычислительной технике наибольшее применение нашли оптические «компакт-диски только для чтения» – CD ROM (Compact Disk Read Only Memory). CD ROM, изготовленные в стандарте ISO/9660, имеют 12 см в диаметре и позволяют хранить до 640 Мегабайт цифровой информации. Сама информация заносится на диск на заводе и далее не может быть изменена.

Отметим ряд существенных достоинств компакт-дисков.

1. При малых физических размерах CD ROM обладают высокой информационной емкостью, что позволяет использовать их в системах мультимедиа, справочных системах со сложной разветвленной структурой данных, учебных комплексах с богатым иллюстративным материалом, в том числе с графическими базами данных. Для наглядности можно отметить, что один компакт-диск CD ROM, имея размеры примерно дискеты, по информационному объему эквивалентен почти пятистам таким дискетам!

2. Считывание информации с компакт-дисков происходит с высокой скоростью, сравнимой со скоростью работы винчестера.

3. Компакт-диски просты и удобны в работе: для смены программного обеспечения достаточно сменить CD ROM (как пластинку в проигрывателе). Перезапись винчестера в таких объемах – достаточно длительная операция, что крайне неудобно при частой смене программного обеспечения, например, от лекции к лекции или от урока к уроку.

4. Компакт-диски практически не изнашиваются. Это свойство имеет особенно большое значение для учебных программ и справочных баз данных, требующих многократных обращений.

5. На компакт-диске невозможно случайно стереть информацию.

6. Компакт-диски не могут быть поражены вирусами.

Для работы с CD ROM требуется незначительная модернизация компьютера, которая заключается в подключении к нему проигрывателя CD ROM (CD ROM Drive), и записи соответствующей программы-драйвера на винчестер. Компакт-диски сменяются как в обычном проигрывателе. Кстати, некоторые проигрыватели CD ROM позволяют прослушивать обычные аудиодиски (CD A), что создает дополнительные удобства для преподавателя, например, иностранного языка. Возможно одновременное подключение к одному компьютеру нескольких проигрывателей CD ROM, что удобно при оснащении информационного центра, например в библиотечном деле. Если учебный класс оборудован локальной сетью, то можно установить проигрыватель CD ROM только на головной машине, при этом остальные пользователи получают доступ к данным по локальной сети. Такое включение устраняет необходимость в установке проигрывателей CD ROM на каждом рабочем месте, а также в закупке большого числа дисков (по числу рабочих мест).

Поскольку CD ROM обеспечивают хранение больших объемов информации, перед разработчиками программного обеспечения открываются богатые возможности в подборе иллюстративного и справочного материала. CD ROM крайне удобны

при использовании больших баз данных, не требующих оперативного изменения.

Что такое проигрыватель CD ROM с технической точки зрения?

Работа этого устройства основана на следующем принципе: участки диска, на которых записаны символы двоичного кода «0» или «1», отличаются коэффициентом отражения лазерного луча, посылаемого проигрывателем CD ROM. Эти отличия улавливаются фотоэлементом и обший сигнал преобразуется в соответствующую последовательность нулей и единиц. Проигрыватель CD ROM проводит первичную обработку считанных сигналов, анализируя и удаляя служебную информацию (проверка позиционирования, контроль и исправление ошибок). Считанные и преобразованные данные проигрыватель CD ROM передает в компьютер через специальную плату адаптера, вставляемую в свободный разъем расширения материнской платы. Конструктивно проигрыватель CD ROM может быть выполнен в виде отдельного устройства (external CD ROM Drive). Его также можно встраивать внутрь корпуса компьютера (internal CD ROM Drive). В последнем случае установочные размеры устройства часто совпадают с дисководом 5.25". Обращение компьютера к CD ROM производится аналогично обращению к другим дискам. Коротко о процессе создания новых дисков.

1. Подготовка исходного программного обеспечения. Главная задача на этом этапе — устранение возможных программных ошибок или несоответствий, так как на готовом CD ROM исправить их будет невозможно.

2. Формирование массива данных для записи на магнитную ленту. При этом формируется «образ» диска, затем весь массив данных, включая программное обеспечение, преобразуется в помехозащищенный код.

3. Изготовление магнитной ленты (мастер-ленты). На специальном устройстве, подключенном к компьютеру, массив данных записывается на магнитную ленту.

Комплекс перечисленных выше операций получил название премастеринг (pre-mastering).

4. Изготовление исходного диска (мастер-диска). Эта операция производится уже на заводе, где данные с мастер-ленты записываются на специальный оптический диск, который в дальнейшем будет служить матрицей для изготовления CD ROM. Данная операция называется мастеринг (mastering).

5. Тиражирование CD ROM, или «репликация» (replication). В заводских условиях с мастер-диска производится копирование информации на CD ROM.

6. Упаковка готовых компакт-дисков и рассылка потребителям.

Сравнительно невысокая стоимость компакт-дисков и проигрывателей CD ROM (от 30 до 100 долларов США за диск и от 500 до 1000 долларов за проигрыватель), а также простота в эксплуатации обусловили их широкое распространение в мире. В пересчете на один бит информации программное обеспечение, поставляемое на CD ROM, в сотни раз дешевле, чем на дискетах. Компании США и Европы (в основном Великобритании) выпустили уже сотни наименований справочных и учебных компакт-дисков — от справочников номеров телефонов и телефаксов до всевозможных иллюстрированных энциклопедий, от курсов иностранных языков до развлекательных и познавательных игр. Персональный компьютер, оснащенный CD ROM проигрывателем, открывает окно в новый информационный мир — мир мульти-медиа.

В России работу по внедрению CD ROM в сферу образования ведет Центр интерактивных средств обучения (ЦИСО) Министерства образования Российской Федерации. На сегодняшний день Центр установил деловые контакты с десятками фирм, производящих CD ROM. Собрана информация и сформирован аннотированный каталог CD ROM, выпускаемых за рубежом. Недавно Центр приступил к созданию отечественного диска.

На основе имеющегося опыта мы дадим рекомендации по выбору, подключению и эксплуатации проигрывателей CD ROM в учебном компьютерном классе, особо обращая внимание на технические и методико-содержательные аспекты.

Каждый из пользователей IBM-совместимых ПК, оснащенных жестким диском, наверняка мечтает о «винчестере» еще большей емкости. Но жесткие диски стоят дорого, и не всем организациям (а тем более индивидуальным пользователям) это по карману. Однако известны программы, позволяющие без каких-либо аппаратных изменений (или с минимальными изменениями) увеличить емкость имеющегося «винчестера» почти вдвое. И одна из таких программ — СТЕКЕР, созданная специалистами фирмы Stac Electronics.

ПРИГЛАШЕНИЕ В СТЕКЕР

Что же такое СТЕКЕР? По сути дела, СТЕКЕР представляет собой архиватор файлов, родственник хорошо известным пользователям архиваторам PKZIP, LHA, LHARC, ARJ и др. Это очень остроумно сделанный архиватор. После запуска ПЭВМ основной модуль рабочего файла загружается в ОЗУ и, оставаясь резидентным, перехватывает все обращения к диску как по командам пользователя, так и программные. А в результате проявляется принципиальное отличие СТЕКЕРА от архиваторов типа PKZIP или LHA: теперь все действия по архивированию/разархивированию файлов производятся автоматически при обращении к диску.

Фактически СТЕКЕР формирует на жестком диске простой архивный файл, автоматически архивируя и добавляя к нему записываемые файлы и разархивируя читаемые. Но для пользователя этот архивный файл «стараниями» СТЕКЕРА выглядит как еще один логический раздел жесткого диска.

СТЕКЕР сам «заботится» о том, чтобы DOS и Norton правильно воспринимали этот «псевдодиск», выдает по запросам этих систем значения емкости «стекерного диска», объема свободного пространства на нем с учетом ожидаемого коэффициента сжатия за счет архивирования и длины хранящихся файлов (значения длины выдаются для разархивированных файлов). Тот же СТЕКЕР следит за всеми обращениями к «стекерному диску», причем отслеживается и большинство способов обращения непосредственно к его «секторам». Так, если мы пытаемся (с помощью, например, функции View/Edit программы PC-Tools) отредактировать содержимое сектора на «стекерном диске», СТЕКЕР разархивирует указанный сектор при чтении и вновь заархивирует его, чтобы переписать поверх старого содержимого. Словом, для нас это будет обычно-

венный жесткий диск, и если не знать заранее, что он порожден СТЕКЕРОм, об этом можно даже и не догадываться.

Полезность СТЕКЕРА можно видеть на таком несложном примере:

- до установки СТЕКЕРА был диск С емкостью 40 Мб;
- при установке СТЕКЕРА на диске С создаются два архивных «стекерных» файла с ожидаемым сжатием 50% и максимальной емкостью по 15 Мб каждый;
- в результате на диске С остаются свободными всего 10 Мб, зато появляются «стекерные диски» D и E емкостью по 30 Мб каждый.

Таким образом, из «винчестера» емкостью 40 Мб мы получили «винчестер» на 70 Мб. Нетрудно подсчитать, что, применив СТЕКЕР, мы получили дополнительно 30 Мб дисковой памяти. И все это чисто программным путем!

Впрочем, СТЕКЕР предусматривает и возможность аппаратной доработки. Если установить на ПК специальную сопроцессорную плату, мы перекладываем «на плечи» стекерного сопроцессора все заботы по архивированию/разархивированию информации и существенно ускоряем выполнение операций чтения/записи на диск, что особенно заметно на компьютерах с небольшим быстродействием.

Количество совместимых со СТЕКЕРОм операционных систем и прикладных программ достаточно велико. Сам же СТЕКЕР умеет гибко подстраиваться под свое окружение. Так, он способен работать с версиями MS-DOS от 3.30 до 5.0 и при этом «старается» использовать по максимуму все предоставляемые операционной системой возможности. Например, на 386 компьютере в MS-DOS 5.0 можно, воспользовавшись встроенными в систему DOS функциями управления расширенной памятью (команда конфигурации

DEVICENHIGH), переместить резидентный модуль СТЕКЕРА в расширенную память, высвободив для основного ОЗУ 54 Кб. Если же у вас нет расширенной памяти, СТЕКЕР позволяет уменьшить объем резидентной части до 18 Кб, пожертвовав, правда, при этом скоростью выполнения операций чтения/записи на диск.

Если вы пользуетесь «услугами» Microsoft Windows, СТЕКЕР прекрасно «уживется» и с этим программным пакетом, а если вы работаете в компьютерной сети, то и здесь СТЕКЕР может предложить вам свои услуги. Однако в обоих последних случаях при работе со СТЕКЕРОм есть ограничения, но они невелики по сравнению с предоставляемыми им удобствами.

Что же касается прикладных программ, то большинство из них (все текстовые редакторы, простые архиваторы, трансляторы с языков высокого уровня, уже упомянутая программа PC-Tools и прочие) даже и не «чувствуют», что работают с «ненастоящим» жестким диском. Лишь бы обращение к ди-

ску производилось в них посредством стандартных функций ДОС.

Некоторые программы, такие, как DISKCOPY (можно разместить «стекерные диски» даже на дискетах, но это требует большей «возни» при их использовании), CHKDSK, FILESAVE из комплекта Norton Utilites или Norton Disk Doktor, также прекрасно ладят со СТЕКЕРОм, хотя в этом случае нужно посмотреть руководство по использованию СТЕКЕРА (если оно есть), чтобы не попасть впросак.

И только немногие программы, отличающиеся специфической реализацией доступа к диску, не могут работать со СТЕКЕРОм. Это часть версий программ хеширования* дисков и оптимизаторы типа COMPRESS или Speed Disk. Однако СТЕКЕР и сам «умеет» хешировать диски, а в комплекте дополнительных утилит, поставляемых вместе с ним, имеется и свой собственный, полностью аналогичный по возможностям названному, оптимизатор «стекерных дисков».

* Хеширование диска — метод реализации доступа к диску, заключающийся в использовании ОЗУ в качестве промежуточного буфера для временного хранения рабочей информации при чтении/записи на диск, что увеличивает скорость выполнения таких операций.

★★★★

ОТВЕЧАЕМ НА "СЕРДИТОЕ ПИСЬМО"

В листинге программы «Четырехголосный музыкальный редактор на Бейсике БК-0010.01» (авторы Г. Пименов, Д. Пименов), опубликованном в ИНФО N2, 1993, в строке 570 по правилам Бейсика не должно быть комментариев в операторе DATA. Публикуя указанный комментарий, редакция надеялась, что работающие с Бейсиком пользователи догадаются, что набирать его не следует. Однако...

Чтобы Бейсик при трансляции программы не выдавал сообщения об ошибке в строке 570, вместо напечатанного следует набрать:

570 DATA 66,84,90,91

Примечание: указанная программа использует для генерации звука встроенный таймер. На тех экземплярах БК-0010.01, в которых встроенный таймер отсутствует, программа может оказаться неработоспособной при правильно набранном тексте.

АЛЬТЕРНАТИВА

ИНФОРМАТИЗАЦИЯ ОБРАЗОВАНИЯ

Комплексные решения в области информатизации образования;

Разработка и поставка прикладного и педагогического программного обеспечения, программно-методических комплексов преподавания информатики;

Большой выбор готовых пакетов программ для КУВТ "Корвет", КУВТ-86, КУВТ-86м, УКНЦ, УКНЦ.01.

ПРЕДСТАВЛЯЕМ НАШИ НОВИНКИ:

Система автоматизации составления расписаний для КУВТ "Корвет";

Инструментальный комплекс программ для младших школьников "Корвет от 2 до 16" для КУВТ "Корвет";

Система обработки и печати текста и графических фрагментов для КУВТ УКНЦ.01 (БК-0011М) с поддержкой принтеров: Epson, MC6313, D100, D100M, Robotron;

Система обмена файлами между компьютерами "Корвет", IBM PC, ДВК и УКНЦ для КУВТ "Корвет";

Локальная вычислительная сеть типа IBM/Корвет" производства фирмы "ЛИнТех" "ITLnet".

Вы можете бесплатно получить новые рекламные материалы и комплект документов для приобретения программных и аппаратных средств, сделав предварительный запрос по телефону:

(385-2) 22 10 50

Позвоните сегодня; если будет занято - перезвоните позже.

МЫ ЖДЕМ ВАШЕГО ЗВОНКА!!!

**ТОО Научно-техническое предприятие "Альтернатива"
г.Барнаул, 656049, а/я 3475, факс (385-2) 23 67 57,
телетайп 233989/І ПКП АЛЬТЕРНАТИВА**

КЛУБ УКНЦ

В.А. Брусенцев

ВИНЧЕСТЕР ДЛЯ УКНЦ

Работая в школе, а затем в институте повышения квалификации учителей на вычислительной технике типа УКНЦ, я постоянно испытывал обычные для учителя информатики трудности с гибкими магнитными дисками: несовместимость дисководов, низкие надежность и быстродействие (особенно заметные при работе с архивами).

Большое облегчение я испытал, когда в компьютерном классе УКНЦ Центра Интерактивных Средств Обучения (ЦИСО), где я теперь работаю, был установлен накопитель на жестких магнитных дисках (НЖМД) – винчестер.

Винчестер подключается к рабочему месту преподавателя подсоединением без каких-либо доработок стандартного адаптера.

Емкость винчестера (20 Мб) позволила разместить на нем все распространенное прикладное и системное программное обеспечение для УКНЦ. Теперь мне не требуется постоянно заниматься переустановкой и перезагрузкой системных дисков. Отдельные программно-методические комплексы размещены в разных разделах винчестера и легко обозримы.

Очень приятной особенностью винчестера, кроме всего прочего, оказалась его высокая надежность. За полгода ежедневной эксплуатации не было ни одного сбоя по вине винчестера.

Таблица демонстрирует сравнительную скорость работы сетевого класса с накопителем на гибких магнитных дисках (НГМД) и с винчестером:

Операция	НГМД	Винчестер
Загрузка ОС на РМП	1 мин	30 с
Загрузка программы размером 64 блока на РМП	40 с	10 с
Разархивация архива размером 198 блоков	10 мин, причем необходима смена дискет	4 мин
Загрузка копии ОС размером 58 блоков сразу на все 12 РМУ	7 мин	5 мин
Одновременная загрузка программы размером 64 блока на все 12 РМУ	11 мин	7 мин
Загрузка копии ОС и программы на одно РМУ после аварии	4 мин	50 с

ЦИСО организует сборку накопителей на жестких магнитных дисках и их поставку в школы с установленным программным обеспечением, включающим в себя известный программно-методический комплекс по информатике «IBM на УКНЦ».

Всегда готов поделиться опытом применения вычислительной техники и программно-методических комплексов в преподавании информатики и других предметов школьного курса. Телефон для справок: 155-87-30.

М.Е. Штарёв

ПРОГРАММА ИССЛЕДОВАНИЯ ФУНКЦИЙ ОДНОЙ ПЕРЕМЕННОЙ

Программа GRAF V 2.1 предназначена для математической обработки и построения графиков функций одной переменной вида $y=f(x)$. С помощью программы можно строить, увеличивать, накладывать в разных масштабах друг на друга любые участки графиков функций, вычислять корни и интегралы, копировать панель графиков функций на принтер и многое другое (см. рис. 1).

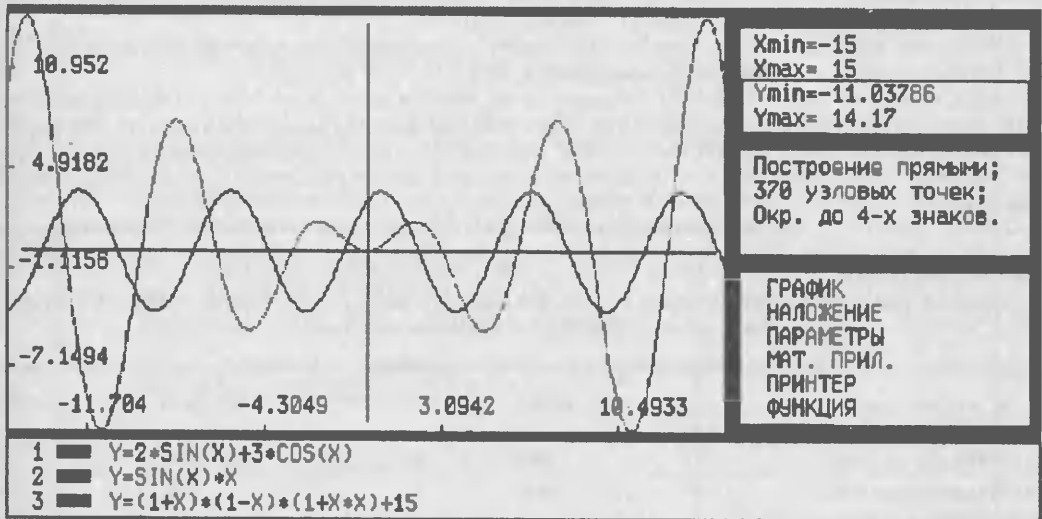


Рис. 1

Несмотря на то, что программа написана на Бейсике, она имеет большие возможности, удобный пользовательский интерфейс, сравнительно высокую скорость работы и другие достоинства. При необходимости «рассматривать под лупой» различные зависимости программа служит в качестве эффективного рабочего инструмента.

Основным недостатком программы (обусловленным Бейсик-системой) является невысокая точность обработки чисел. Так, расчет корней возможен с точностью до пяти знаков после запятой.

Необходимо отметить, что автор не случайно выбрал именно Бейсик. Во-первых, программы, написанные на этом языке,

каждый желающий может доработать в соответствии со своими вкусами. А во-вторых, у Бейсика есть много возможностей, трудно реализуемых на других языках.

Для работы программы требуется файл текущих значений (GRAF21.TMP), изначально совпадающий со строками программы 1-9. Все переменные этого файла полностью доступны из программы.

Программа написана на машинно-независимом языке, ее несложно перенести на другой тип компьютера. Для этого следует обратить внимание на адаптацию графики, подпрограмму вывода на принтер и убрать «музыкальную часть» (строки 11-19, 270-273).

ОПИСАНИЕ ПРОГРАММЫ

Программа имеет шесть панелей: панель графиков функций, текущих координат, отображения режимов работы, меню текущих функций и вспомогательная панель. Пользователь отдает приказы в основном с помощью разветвленной системы меню, на которую мы и будем опираться.

Основное меню

- Пункт «ГРАФИК»: в этом пункте производится построение графика с предварительной очисткой панели графиков функций, перерасчетом осей координат, оцифровки этих осей в соответствии с текущими масштабами. График строится (как и в пункте «Наложение») 8-м, 6-м или 4-м цветом в соответствии с номером выбранной функции.
- Пункт «НАЛОЖЕНИЕ»: в этом пункте можно наложить график выбранной функции на графики функций, построенных раньше. Оси координат и оцифровка не перестраиваются.

При работе с этим пунктом надо быть внимательным, т. к. график накладывается в соответствии с текущими масштабами, в то время как оси координат могли быть построены в других масштабах. Благодаря этому эффекту программа оказывается более гибкой по возможностям.

- Пункт «ПАРАМЕТРЫ» является входом во второе меню.
- Пункт «МАТ. ПРИЛ.»: здесь можно провести некоторую математическую обработку зависимостей (см. «Меню математических приложений»).
- Пункт «ПРИНТЕР» позволяет (с помощью вспомогательного меню) копировать панель графиков функций на принтере. В стандартном листе «А4» в ширину умещается две панели и поэтому выводить их можно в две колонки. Программа отлажена для принтера

«МС-6312» и работает с EPSON-совместимыми принтерами.

- Пункт «ФУНКЦИЯ» позволяет менять обрабатываемые функции не выходя из программы (см. меню «Функция»).

Меню параметров

- Пункт «Xmin/Xmax» позволяет установить Xmin и Xmax для последующего построения функций. С помощью этого пункта можно увеличивать или уменьшать график функции по горизонтали, перемещать его по панели в горизонтальном направлении. При этом выбранные Xmin и Xmax отобразятся на панели текущих координат. Выбирая горизонтальные координаты, надо следить, чтобы в промежутке между ними функция была определена. Если в этом промежутке имеются только «выколотые» точки, то обойти указанное ограничение можно, манипулируя количеством узловых точек.
- Пункт «Ymin/Ymax» выводит в меню третьего уровня вложения. Среди пунктов этого меню:
 - Пункт «Установка» позволяет установить желаемые Ymin и Ymax. При этом новые значения отобразятся на панели текущих координат.
 - Пункт «Сброс» перекладывает проблему установки Ymin и Ymax для построения графиков функций на машинное исполнение. В этом режиме перед каждым построением (не наложением!) графика функции машина рассчитывает оптимальные Ymin и Ymax и выведет их на панель текущих координат. Индикатором этого режима работы является разделительная полоса между горизонтальными и вертикальными координатами на панели текущих координат.

Надо отметить, что при использовании пункта «ГРАФИК» оси координат всегда располагаются так, что точка их пересечения указывает на начало координат.

Пункт «Количество точек» служит для задания количества узловых точек для построения графиков. Чем больше количество узловых точек, тем выше качество построения графика функции, но вместе с тем и дольше производится это построение. Количество узловых точек отображается на панели режимов работы и может находиться в пределах от 1 до 600. При задании можно использовать «забой».

С помощью пункта «Способ вывода» можно выбрать режим построения графика: точками или линиями. Последнее удобно использовать, если при построении графика точками они располагаются неравномерно.

В пункте «Округление» с помощью подменю можно задать степень округления выводимых чисел при оцифровке осей и некоторых других операциях.

Меню «функция»

В пунктах «Функция 1» – «Функция 3» можно задать желаемые функции для дальнейшей обработки. В задании обрабатываемых функций можно использовать все стандартные функции Бейсик-системы и ряд дополнительных. Программа дает пользователю возможность определять свои, наиболее часто употребляемые функции, со строки 24 (например: 24 DEF FNC(X)=1/TAN(X)). Ссылки на эти, а также на любую из трех, обрабатываемых в данный момент зависимостей, можно использовать в задании функций для обработки.

Дополнительно определены:

$FNL(A, X)$ – логарифм X по основанию A ;

$FNO(A, X)$ – результат X , округленный до A знаков после запятой.

После ввода функции на панели текущих функций появится соответствующий

флажок. Уже переопределенные функции (у которых есть флажок) можно переопределять заново. Для перевода программы на нововведенные функции надо выбрать в подменю пункт «ВВОД». К обрабатываемым зависимостям можно обращаться по именам $FNF1(X)$ – $FNF3(X)$.

Примеры задания функций:

$X^*X^*(2-SIN(X))+FNF1(X)*COS(FNF2(X))$ – (для 3-ей функции),
 $FNO(2, RND(X)*10)$.

После того как интересующие функции набраны, нужно пунктом «ВВОД» перенастроить программу на работу с ними.

Меню «мат. прил.»

Пункт «Корни» позволяет рассчитать корни исследуемых функций. Выбирать интервал для поиска корней желательно таким, чтобы функция была на нем монотонна и имела только один корень. В принципе, программа ищет на обозначенном участке такой X , для которого $|F(X)|=min$, что можно использовать в некоторых случаях для нахождения локальных экстремумов на данном интервале.

В пункте «Интеграл» можно найти определенный интеграл функции на заданном участке методом трапеций.

ПРИМЕЧАНИЕ. Программа требовательна к памяти, и поэтому перед ее запуском освободите ОЗУ от лишних драйверов, разрешите свопинг USR и использование ДОС, отводящей наибольшее место в ОЗУ для программ пользователя.

По той же причине в текст не следует вносить комментарии, «расшифровывать» операторы и вставлять пробелы туда, где их нет.

1 DEF FNF1(X)=SIN(2*X)+COS(X)+PI
 2 DEF FNF2(X)=X/((X-2)*(X-2))

```

3   DEF FNF3(X)=SIN(2*X)+COS(X)+PI-X/
   ((X-2)*(X-2))
4   RS%= 2
5   XN= 0
6   XM= 5
7   YN=-1.94
8   YM=-9.696152E-03
9   OK= 4
10  DIM NFX(3),Y(601),T%(6)
11  DATA 4096,26,-50,90,1,0,80,-1,4983,
   -10,4134,4198,5568,92,5569,5,258,
   -27617,-578,-29729,-580,-32515,32326,
   5505,5504,135,30592,31616,29056,
   32128,30848,32384
12  FOR AD%=80TO130ST2
13  READ A%
14  POKE AD%,A%
15  NEXT
16  DEF USR=96
17  FOR I%=1TO6
18  READ T%(I%)
19  NEXT
20  KT%=200
21  DEF FNL(A,X)=LOG(X)/LOG(A)
22  DEF FNO(A,X)=INT(X*10^A+
   .555555555)/10^A
23 '
24 '
25 '
26  SAVE "GRAF21"
27  OPEN "GRAF21.TMP" FOR INPUT
28  FOR I%=1TO3
29  INPUT # A$
30  NFX(I%)=MID$(A$,INSTR(A$,"")+1)
31  NEXT
32  CLOSE
33  WIDTH 80
34  COLOR 8,1,1
35  SCREEN 2
36  LINE (0,70)-(447,260),,B
37  LINE (0,67)-(639,31),,B
38  LINE (455,74)-(633,145),,B
39  LINE (455,152)-(633,200),,B
40  LINE (455,206)-(633,255),,B
41  ? AT(22,7)"GRAF V 2.1"AT(4,16)
   "Штапов М.Е.      г. Москва
   20.02.1993г."
42  PAINT (452,75),5,8
43  GOSUB 313
44  FOR I%=1TO3
45  ? AT(1,17+I%)I%" Y="NFX(I%)
46  LINE (32,74-I%*11)-(52,69-I%*11),
   10-I%*2,BF
47  NEXT
48  GOSUB 281
49  MN%=1
50  GOSUB 249
51  IF A<>3TH 95
52  MN%=2
53  GOSUB 249
54  IF A=4TH?AT(2,22)"Количество
   точек=";EL59
55  GOSUB 290
56  KT%=J
57  GOSUB 288
58  GOTO 75
59  IF A<>3TH73 ELMN%=3
60  GOSUB 249
61  F A=2THRS%=-ABS(RS%)EL71
62  ? AT(2,22)"Ymin=";
63  GOSUB 290
64  YN=J
65  ? AT(2,23)"Ymax=";
66  GOSUB 290
67  YM=J
68  GOSUB 288
69  IF YN>=YM TH62 ELGOS281
70  GOTO 60
71  IF A=3THRS%=ABS(RS%)EL52
72  GOTO 69
73  IF A<>5TH77
74  IF ABS(RS%)=1THRS%=2*RS%ELRS%
   =RS%/2
75  GOSUB 313
76  GOTO 52
77  IF A=2 TH?AT(2,22)"Xmin=";EL86
78  GOSUB 290
79  XN=J
80  ? AT(2,23)"Xmax=";
81  GOSUB 290
82  XM=J
83  GOSUB 288
84  IF XN>=XM TH77 ELGOS281
85  GOTO 53

```

```

86 IF A=1TH49 ELMN%=6
87 GOSUB 249
88 IF A=1TH52
89 IF A=2THOK=0ELOK=1
90 IF A=4THOK=2
91 IF A=5THOK=4
92 IF A=6THOK=-1
93 GOSUB 313
94 GOTO 87
95 IF A<>1ANDA<>2TH143 ELGOS302
96 XR=XM-XN
97 XS=XR/KT%
98 IF A=2TH129
99 IF RS%<0TH110
100 K%=1%
101 FOR X=XN TOXM STXS
102 GOSUB 309
103 Y(K%)=Y
104 IF K%=1%THYN=Y EL106
105 YM=Y
106 IF Y>YM THYM=Y
107 IF Y<YN THYN=Y
108 K%=K%+1
109 NEXT
110 LINE (1,71)-(446,259),1,BF
111 GOSUB 281
112 RX=446/XR
113 YR=YM-YN
114 RY=188/YR
115 XP=1-XN*RX
116 YP=71-YN*RY
117 FOR I=0TO3
118 XT=50+I*110
119 YT=100+I*45
120 ? AT(XT/8-2,16);
121 IF OK=>0TH?FNO(OK,(XT-XP)/RX)EL?(
XT-XP)/RX
122 ? AT(1,24-YT/11);
123 IF OK=>0TH?FNO(OK,(YT-YP)/RY)EL?
(YT-YP)/RY
124 LINE (XT,70)-(XT,73)
125 LINE (0,YT)-(4,YT)
126 NEXT
127 IF ABS(XP)<447THLIN(XP,75)-(XP,255)
128 IF YP<260AND YP>70THLIN(10,YP)-
(440,YP)
129 K%=1
130 J%=0
131 FOR X=XN TOXM STXS
132 XT=X*RX+XP
133 IF A=2 OR RS%<0THGOS309 ELY=Y(K%)
134 YT=Y*RY+YP
135 IF YT<71 OR YT>259TH139
136 IF ABS(RS%)=1% OR J%=0% TH
PSET(XT,YT),C% EL LIN-(XT,YT),C%
137 J%=1%
138 GOTO 140
139 J%=0%
140 K%=K%+1%
141 NEXT
142 GOTO 49
143 IF A=4THMN%=7EL183
144 GOSUB 249
145 IF A=1TH49 ELGOS302
146 ? AT(2,22)"Xmin?";
147 GOSUB 290
148 XI=J
149 ? AT(2,23)"Xmax?";
150 GOSUB 290
151 XA=J
152 GOSUB 288
153 IF A=2 TH166 ELDX=(XA-XI)/400
154 X=XI
155 GOSUB 309
156 S=Y/2
157 FOR X=XI+DX TOXA-DX STDX
158 GOSUB 309
159 S=S+Y
160 NEXT
161 X=XA
162 GOSUB 309
163 S=(S+Y/2)*DX
164 ? AT(2,22)"Определенный
интеграл="S,""FNO(2,S);
165 GOTO 180
166 B=XA-XI
167 IF ABS(B)<5E-7TH178
168 A=B/2+XI
169 X=(XA+A)/2
170 GOSUB 309
171 X1=ABS(Y)
172 X=(X1+A)/2
173 GOSUB 309
174 IF X1<ABS(Y)THX1=A ELXA=A

```

```

175 IF E=B TH177 ELE=B
176 GOTO 166
177 ? AT(2,22)"Достигнута точность="
CHR$(7)B
178 ? AT(2,23)"X="XA,"";
179 ? FNO(2,XA);
180 ? ")","Press key.";
181 IF INKEY$="TH181 ELGOS288
182 GOTO 144
183 IF A=5THMN%=4EL210
184 GOSUB 249
185 J%=0
186 IF A=3THI%=1EL190
187 FOR J%=1TO21
188 LPRINT CHR$(27)"J"@";
189 NEXT
190 IF A=1TH49 ELA%=-32472
191 LPRINT CHR$(27)"A"CHR$(8);
192 FOR X%=1TO56
193 A%=A%-1%
194 B%=A%
195 LPRINT CHR$(10);STRING$
(40*I%,32)CHR$(27)"K";
196 C%=191%
197 GOSUB 208
198 LPRINT CHR$(0);
199 FOR Y%=0TO190
200 POKE -608%,B%
201 C%=PEEK(-606%)
202 C%=INT(C%/256%) OR (C% AND
255%)
203 GOSUB 208
204 B%=B%+80%
205 NEXT Y%,X%
206 LPRINT CHR$(27)"A"CHR$(12);
207 GOTO 184
208 IF INP(-588%,128%)=128%THPOK
-586%,C%EL208
209 RETURN
210 MN%=5
211 GOSUB 256
212 IF A=1TH49
213 IF A=5TH237 ELB$=""
214 A=A-1
215 ? AT(2,22)"A"ФУНКЦИЯ: Y=";
216 I%=1
217 A$=INKEY$
218 IF A$=""TH217
219 IF A$=CHR$(13)TH233
220 IF I%<63 OR A$=CHR$(127)TH223
221 BEEP
222 GOTO 217
223 IF A$<>CHR$(127)TH229
224 IF I%=1TH221 ELB$=MID$(B$,1,I%-2%)
225 LINE (LEN(B$)*8+128,13)-(639,
25),1,BF
226 I%=I%-1%
227 ? AT(16+LEN(B$),22);
228 GOTO 217
229 B$=B$+A$
230 I%=I%+1%
231 ? A$;
232 GOTO 217
233 NF$(A)=B$
234 LINE (8,70-A*11)-(11,72-A*11),8,BF
235 GOSUB 288
236 GOTO 210
237 OPEN "GRAF21.TMP"FOR OUTPUT
238 ? # "1 DEF FNF1(X)="NF$(1)
239 ? # "2 DEF FNF2(X)="NF$(2)
240 ? # "3 DEF FNF3(X)="NF$(3)
241 ? # "4 RS%="RS%
242 ? # "5 XN="XN
243 ? # "6 XM="XM
244 ? # "7 YN="YN
245 ? # "8 YM="YM
246 ? # "9 OK="OK
247 CLOSE
248 MERGE "GRAF21.TMP",R
249 DATA 6,ГРАФИК,НАЛОЖЕНИЕ,
ПАРАМЕТРЫ,МАТ. ПРИЛ.,ПРИНТЕР,
ФУНКЦИЯ
250 DATA 6,ВЫХОД,Xmin/Xmax,
Ymin/Ymax,Кол-во точек,Способ
вывода,Округление
251 DATA 3,ВЫХОД,Установка,Сброс
252 DATA 3,ВЫХОД,Первая колонка,
Вторая колонка
253 DATA 5,ВЫХОД,Функция 1,Функция
2,Функция 3,Ввод
254 DATA 6,ВЫХОД,До целого,До 1
знака,До 2 знаков,До 4 знаков,
Округл. выкл.
255 DATA 3,ВЫХОД,Корни,Интеграл

```

```

256 LINE (456,144)-(632,75),1,BF
257 IF MN%=1THRES249 ELRES250
258 IF MN%=3THRES251
259 IF MN%=4THRES252
260 IF MN%=5THRES253
261 IF MN%=6THRES254
262 IF MN%=7THRES255
263 A=1
264 READ J
265 FOR I=1TOJ
266 READ A$
267 ? AT(59,10+I)A$
268 NEXT
269 LINE (460,154-A*11)-(600,143-A*11),,B
270 I%=USR(-T%(A))
271 FOR I%=1TO1000
272 NEXT
273 I%=USR(-32768)
274 A$=INKEY$
275 IF A$=""TH274
276 LINE (460,154-A*11)-(600,143-
A*11),1,B
277 IF ASC(A$)=13THENRETELA$=INKEY$
278 IF A$="A"THIFA=1THA=J ELA=A-1
279 IF A$="B"THIFA=J THA=1ELA=A+1
280 GOTO 269
281 LINE (456,253)-(632,207),1,BF
282 ? AT(58,1)"Xmin="XM
283 ? AT(58,2)"Xmax="XM
284 ? AT(58,4)"Ymax="YM
285 ? AT(58,3)"Ymin="YN
286 IF RS%>0 TH LIN(456,231)-
(632,207),,B
287 RETURN
288 LINE (0,0)-(639,30),1,BF
289 RETURN
290 B$=""
291 A$=INKEY$
292 IF A$=""TH291 ELB=ASC(A$)
293 IF B=13TH297
294 IF B=127 AND B$<>""TH299
295 IF B<45 OR B>57TH291 ELB$=B$+A$
296 GOTO 300
297 IF MID$(B$,1,1)="-"THJ=
-VAL(MID$(B$,2))ELJ=VAL(B$)
298 RETURN
299 IF LEN(B$)=1THB$=""ELB$=
MID$(B$,1,LEN(B$)-1)
300 ? A$;
301 GOTO 291
302 ? AT(2,22)"Номер функции:";
303 A$=INKEY$
304 J=VAL(A$)
305 IF J<1 OR J>3TH303 ELNF%=J
306 C%=10-NF%^2
307 GOSUB 288
308 RETURN
309 IF NF%=1%THY=FNF1(X)
310 IF NF%=2%THY=FNF2(X)
311 IF NF%=3%THY=FNF3(X)
312 RETURN
313 ? AT(58,6)"Построение ";
314 IF ABS(RS%)=1 TH?"точками;"EL
?"прямыми;"
315 ? AT(57,7)KT%"узловых точек; "
316 ? AT(58,8)"Окр. ";
317 IF OK=-1TH?"выключено. "
318 IF OK=0TH?"до целого. "
319 IF OK=1TH?"до 1-го знака."
320 IF OK=2TH?"до 2 знаков."
321 IF OK=4TH?"до 4 знаков."
322 RETURN

```

КЛУБ «КОРВЕТ»

В.Р. Степанов, Е.Н. Кремчуков

ИСПОЛЬЗОВАНИЕ СПРАЙТОВ В БЕЙСИК-ПРОГРАММАХ НА «КОРВЕТЕ»

Графическое изображение строится различными способами. Рисовать картинку на экране можно с помощью графических операторов Бейсика или используя библиотеки Си, Паскаля, EXPRESS-PASKAL. Но эти методы не всегда устраивают.

Бейсик имеет малое быстродействие, а программирование на Си и Паскале затруднительно для начинающих программистов. Кроме того, эти компиляторы включают библиотеки, что заметно увеличивает объем программы.

Авторы данной статьи видят выход в методе построения изображений с помощью ассемблерных вставок в Бейсик-программы. Такой способ наиболее удобен, так как сочетает простоту создания программ, присущую Бейсику, с быстродействием ассемблера. В частности, таким образом с помощью операторов Бейсика можно создавать массивы изображений, не требующих высокого быстродействия и состоящих из простых фигур, например рамки поля. С помощью же ассемблерных вставок можно строить спрайты, причем любых размеров (разумеется, в пределах графического экрана).

В общем случае алгоритм рисования выглядит следующим образом. Допустим, вам понадобилось нарисовать картинку на бумаге. Что вы для этого сделаете? Вы возьмете: (1) — лист чистой бумаги; (2) — набор имеющихся у вас красок, т.е. выберете палитру; (3) — выберете нужный

цвет и (4) — начнете рисовать, нанося мазки краски на лист бумаги. Мы специально выделили последовательность действий, потому что «Корвет» для того, чтобы что-то нарисовать, поступает таким же образом.

Для вывода графических изображений на экран используется специально выделенная область памяти, называемая графическим запоминающим устройством (ГЗУ), или видеопамятью. ГЗУ как бы проецирует свое содержимое на экран дисплея; одной ячейке видеопамати соответствует один элемент экрана.

Существует несколько способов вывода информации на экран. В одном из них за каждую точку экрана отвечает одна ячейка ГЗУ, в которой хранится информация о цвете точки, интенсивности ее свечения, а также о том, зажжена она или нет. В другом способе цвет и интенсивность свечения точки определяются заранее перед процедурой рисования, а каждая ячейка ГЗУ сообщает о включении целой группы точек. Образно говоря, если в первом случае рисунок состоял бы из точек, то во втором — из штрихов. У каждого способа есть свои достоинства и недостатки; не будем здесь о них говорить, отметим только, что «Корвет» рисует, используя второй способ.

Наш компьютер применяет байтовую графику, т.е. при одном обращении к ячейке ГЗУ он выводит сразу 8 точек одно-

го цвета. Давайте подсчитаем, какой объем оперативной памяти понадобится для этого. Как известно, «Корвет» имеет графику высокого разрешения — 512·256 точек. Если использовать первый метод рисования, то понадобится $512 \cdot 256 = 131072$ байт памяти, или 128 Кбайт. В нашем случае при рисовании одновременно 8 точек понадобится всего $131072/8 = 16384$ байт, или 16 Кбайт. Именно такой объем имеет включаемый в адресное пространство компьютера сегмент ГЗУ.

Но самое интересное на этом не заканчивается. Наиболее распространенный вариант компьютера имеет ГЗУ объемом 48 Кбайт. Оно разделено на три плоскости объемом по 16 Кбайт. При начальной инициализации этим плоскостям присваиваются физические цвета, но в процессе работы программы вы можете их перепрограммировать. Программист может использовать для вывода графики как одну плоскость, так и все сразу. В первом случае режим работы с ГЗУ называется послыйным, а во втором — цветовым. При работе в послыном режиме процессор обращается только к текущей плоскости, заданной маской регистра цвета, т. е. не к 48, а только к 16 Кбайт. При использовании цветового режима происходит наложение плоскостей, и суммарный объем ГЗУ составит также 16 Кбайт, которые включаются как сегмент ОЗУ в адресное пространство компьютера в конфигурациях ROMB2, BASG, DOSG1, DOSG2.

Но и это еще не все. Авторская концепция подразумевает существование четырех независимых границ ГЗУ, которые практически мгновенно могут переключаться. Используя этот принцип, можно очень легко реализовать мультипликацию. Чтобы не обольщать читателей, надо сразу сказать, что в компьютерах производства г. г. Баку, Минск, С.-Петербург, Каменск-Уральский реализована только одна страница, а значит, описанный выше подход неприемлем.

ГЗУ может быть включено в карту памяти либо с адреса 0C000H по 0FFFFH, либо с 4000H по 8000H. Программист сам должен выбрать нужную конфигурацию

для решения своих задач. Например, для использования графических функций на ПК8010 (конфигурация BASIC) рекомендуется использовать BASG. При использовании ОС CP/M или МикроDOS лучше применять DOSG1, DOSG2. Наиболее привлекательной является конфигурация ROMB2, которая, кроме ОЗУ, содержит набор драйверов, АЦЗУ, ГЗУ, т. е. оптимальный набор компонентов для работы с дисплеем. Эта конфигурация позволяет достаточно просто совмещать текстовый и графический экраны.

Из сказанного следует, что перед тем как начать процедуру рисования, необходимо переключиться в конфигурацию, содержащую ГЗУ. Сделать это можно, воспользовавшись «услугами» системного регистра. Но прежде нужно обратить внимание на следующий момент: при переключении конфигурации памяти следует специальной командой процессора запретить прерывания, иначе можно увидеть очень интересное изображение на экране, неизбежно заканчивающееся появлением девяток, и, увы, совсем не то, которое задумывалось.

Перейдем к определению цвета рисования. Цвета определяет другая специальная ячейка памяти — регистр цвета. Он находится на той же странице, что и системный регистр, но его относительный адрес равен 0BFH. Например, для конфигурации BASG адрес регистра цвета будет составлять 0BFBFH, а для ROMB2 — 3ABFH. С помощью регистра цвета можно также задавать режим рисования — цветовой или послыйный. Для создания процедуры рисования спрайтов нас будет интересовать цветовой режим записи в ГЗУ.

Рассмотрим подробнее формат регистра цвета. В нем каждый бит несет свою смысловую нагрузку.

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

- D1-D3 — определяют плоскости для чтения в ГЗУ;
- D4-D6 — определяют плоскости для записи в ГЗУ;
- D7 — режим работы.

При работе в цветовом режиме бит D7 равен 1, а биты D3 – D1 указывают, какие плоскости будут участвовать в процессе рисования. Содержимое битов D6 – D4 и D0 в данном случае значения не имеет. Комбинация битов D3 – D1 будет определять цвет записываемого байта. Здесь нужно учесть, что таким образом будет присвоен логический (т. е. условный) цвет, а физический определяется с учетом программирования просмотрной таблицы. Допустим, вы хотите нарисовать байт логического цвета 5 (если таблицу цветов перед рисованием не перепрограммировали, то логический цвет будет соответствовать физическому, т. е. фиолетовому). В этом случае в регистр цвета вам нужно записать двоичное число 1000101х, где содержимое бита, обозначенного как "х", не имеет значения (в шестнадцатиричном виде – 8AH или 8BH, в зависимости от значения бита D0).

Теперь вы знаете, как выбрать «лист бумаги» и подготовить «карандаш» нужного цвета; остается научиться собственноручно рисовать. Как это сделать? После вышеуказанных действий вы должны заносить в ГЗУ байты, соответствующие выбранному вами рисунку и месту на экране дисплея. Покажем, как это делается. Возьмем, к примеру, конфигурацию BASG, содержащую ГЗУ, которое начинается с адреса 0C000H и занимает область до адреса 0FFFFH. Байт 0FFH, записанный по адресу 0C000H, вызовет появление на экране линии, состоящей из восьми точек и расположенной в левом верхнем углу, т. е. соответствует оператору Бейсика LINE (0,0)-(8,0). Так как по горизонтали «Корвет» рисует 512 точек, то это составит $512/8=64$ байт, т. е. одна строка состоит из 64 байт (40H). Значит, адрес 0C03FH будет соответствовать отрезку, находящемуся в правом верхнем углу, т. е. LINE (504,0)-(512,0).

Читатель теперь, вероятно, вычислит сам, в каком месте экрана будет находиться отрезок, соответствующий адресу 0FFFFH.

Теперь можно перейти к созданию процедуры рисования спрайтов. Предложенная авторами методика позволяет со-

здавать спрайты любых размеров в пределах графического экрана.

Представленная процедура рисования состоит из двух частей: составления самой программы рисования и формирования блока данных. Причем сама программа рисования будет фактически одинаковой для любых спрайтов. В ней будут меняться только некоторые параметры (цвет, размеры по горизонтали, по вертикали, месторасположение). Значительно изменяться будет только блок данных. Ниже мы покажем, как его составлять. Пусть нам потребовалось нарисовать яблоко. Прежде чем рисовать его на экране, сделаем макет на бумаге в клетку. Число клеток по горизонтали должно быть кратно 8, т. е. 8, 16, 24 и т. д. Если это не получается, необходимо сделать дополнение до 8 по горизонтали. Например, если рисунок яблока занял по горизонтали 14 клеток, для создания спрайта надо дополнить его до 16, т. е. пририсовать к левому (или правому) краю 2 клетки. Это делается для создания байтов блока данных (8 последовательно расположенных клеток составят один байт).

Клеткам, занимаемым рисунком, присвоим значение "1", а свободным – значение "0", таким образом мы получим массив двоичных чисел, которые легко можно преобразовать в набор байтов. В нашем случае это преобразование будет выглядеть следующим образом.

Двоичное выражение	Шестнадцатиричное выражение
00000100	04 00
00000010	02 00
00111010	3A F0
01111111	7F F8
11111111	FF FC
11111111	FF FC
11111111	FF FC
11111111	FF FC
11111111	FF FC
11111111	FF F8
01111111	7F F8
01111111	7F F0
00111111	3F E0
00011001	19 C0

Мы получили блок данных для рисунка. Нужно отметить, что при создании спрайта лучше пользоваться стандартными шаблонами, причем размер по вертикали (количество строчек) может быть лю-

бым. Теперь, имея блок данных, можно написать полную программу рисования яблока. Тут для удобства работы желательно использовать компилятор ассемблера. Итак, для спрайта «яблоко» имеем:

```

LEN :      EQU 2H          ; LEN — значение длины спрайта по горизонтали,
HIGH:      EQU 0DH        ; равно количеству байтов
PLACE:     EQU 0DD20H     ; HIGH — количество строк в спрайте,
                                ; т.е. размер по вертикали
                                ; PLACE — адрес места спрайта на экране
MVI A, 6CH          ; Переключение в конфигурацию DOSG2
STA 0FA7FH          ; Выбор цвета 8FH — белый цвет
MVI A, 8FH
STA 0BFBFH
LXI H, PLACE
LXI D, DATA        ; Основная процедура рисования
MVI C, HIGH         ; В регистре C — счетчик числа строк
PUSH H             ; Сохраняем начальную координату в стеке
STRING:      MVI B, LEN   ; В регистре B — счетчик числа байтов
LOOP:        LDAX D       ; В строке
MOV M, A           ; Далее идет стандартная процедура
INX H              ; копирования
DCR B              ;
NZ LOOP           ; Строка закончилась ?
POP H              ; Если да, то переход на следующую. Для
PUSH D             ; этого извлекается из стека адрес начала
LXI D, 0040H      ; строки и прибавляется 64 (40H),
DAD D              ; потом все восстанавливается
PUSH H
POP D
DCR C
JNZ STRING        ; Строчки закончились ?
HLT               ; Если да, то останов
DATA:        DB 04, 00, 02, 00, 3AH, 0F0H, 7FH, 0F8H, 0FFH, 0FCH, 0FFH, 0FCH
DB 0FFH, 0FCH, 0FFH, 0FCH, 0FFH, 0F0H, 7FH, 0F8H, 7FH, 0F0H, 3FH
DB 0E0H, 19H, 0C0H
END

```

Данная программа занимает 74 байт и будет выполнена за 1 мс (одна тысячная доля секунды). Авторы создавали спрайты размером 128·100, при этом изображение появлялось фактически мгновенно. Кстати, можно модифицировать программу, уменьшив ее размер и время выполнения. Предоставим читателям самим поэкспериментировать в этом направлении.

Как вы уже заметили, данная программа легко настраивается на любые параметры спрайта. Изменением константы LEN можно изменять размер по горизонтали, а HIGH — по вертикали. Изменяя начальный адрес PLACE, можно пере-

мещать спрайт. Можно также использовать другой цвет, меняя константу записи в регистре цвета.

А теперь проследим, как делается ассемблерная вставка в Бейсике. Для вызова подпрограмм на машинном языке служит оператор USR. Он имеет формат X=USR[n](адрес), где X — фиктивная переменная, а n — номер подпрограммы, определенный оператором DEFUSR[n]. Чтобы запустить программу на машинном языке, ее нужно разместить в памяти. Далее все примеры будут приводиться для интерпретатора Бейсик версии 1.0, рабо-

тающей под управлением ОС CP/M. Эта программа работает и на МикроDOS, и на РМУ, только в этом случае используемые

конфигурации и значения системного регистра должны быть другими.

Итак, машинные коды приведенной выше программы после создания командного файла будут иметь следующий вид (без блока данных):

```
0100 F3 3E 6C 32 7F FA 3E 8F 32 BF BF 0E 0E 21 20 DD
0110 11 33 01 E5 06 02 1A 77 23 13 05 C2 17 01 E1 D5
0120 11 40 00 19 D1 E5 0D C2 15 01 3E 1C 32 BF BF FB
0130 C9
```

Получить листинг можно, воспользовавшись программами DDT, PKTOOL или POWER. Добавим блок данных:

```
0130 C9 04 00 02 00 3A F0 7F F8 FF FC FF FC FF FC FF FC
0140 FF F0 7F F8 7F F0 3F E0 19 C0
```

Это и есть наша подпрограмма рисования спрайта. Напишем теперь программу на Бейсике, использующую эту подпрограмму.

```
10 'Размещение подпрограммы в памяти
15 CLS: CLEAR: FOR I=0 TO 74: READ C$: POKE &HBD00+I, VAL("&H+C$"): NEXT I
20 'Определение и вызов подпрограммы
25 DEFUSR=&HBD00:X=USR(0)
30 DATA F3,3E,6C,32,7F,FA,3E, BF, 32,BF,BF,0E,0E,21,20,DD
35 DATA 11,33,BD,E5,06,02,1A,77,23,13,05,C2,17,BD,E1,D5
40 DATA 11,40,00,19,D1,E5,0D,C2,15,BD,3E,1C,32,7F,BF,FB
45 DATA C9
50 'Блок данных
55 DATA 04,00,02,00,3A,F0,7F,F8,FF,FC,FF,FC,,,FF,FC
60 DATA FF,F0,7F,F8,7F,F0,3F,E0,19,C0
```

Авторы намеренно выделили некоторые ячейки памяти в размещаемой подпрограмме. Изменяя содержимое этих ячеек, можно добиться нужных вам результатов. И еще, как вы заметили, мы разместили подпрограмму, начиная с адреса &HBD00, и поэтому в программе поменялись адреса переходов (если в дампе памяти значение перехода равно 0115, то в последнем случае будет BD15 и т.д.). Пользователь может разместить программу в любом месте ОЗУ, кроме области, занимаемой интерпретатором и ОС. Но

авторы советуют размещать подпрограмму в самой верхней доступной пользователю части ОЗУ.

Рассмотрим ячейки памяти, которые можно изменять в вышеприведенной программе:

- &HBD07 – цвет ячеек;
- &HBD0C – количество строчек, вертикальный размер;
- &HBD0E, &HBD0F – месторасположение спрайта на экране;
- &HBD15 – количество байт в строчке, горизонтальный размер.

В качестве демонстрации покажем, как организовать перемещение спрайта по экрану. Сделать это можно, изменяя содержимое ячеек &HBD0E и &HBD0F. Простейшее горизонтальное передвижение спрайта реализуется так: надо изменить строку 25 приведенной Бейсик-программы и ввести еще одну строку.

Опыт авторов в создании и использовании спрайтов показывает, что наиболее трудоемка операция составления блока

```
25 DEFUSR = &HBD00:L=&HDD
26 PCLS:I=I+1:POKE&HBD0E, &H20+I:
   X=USR(0):IFPEEK(&HBD0E)=&HFF
   THENPOKE&HBD0E,0:POKE&HBD0F,I+1
27 GOTO26
```

КОМПЬЮТЕРНАЯ ГРАФИКА

В Корнеллском университете (Итака, шт. Нью-Йорк) разработан машинный способ получения движущегося графического изображения, который позволит значительно усовершенствовать современную технику мультипликации.

Новый способ основан на представлении, что объекты являются твердыми телами и не могут проникать друг в друга. Перемещение объектов определяется ньютоновскими законами движения, а для расчета сил взаимодействия между соприкасающимися или сталкивающимися объектами используется специально разработанный математический аппарат.

В соответствии с процедурами, заключаемыми в составляемую по новому способу моделирующую программу, перемещение вводимых в изображение объектов рассчитывается для каждого данного момента с определением их возможного взаимного отскокивания, откатывания, проскальзывания и т.п. При этом силы взаимодействия рассчитываются с высокой точностью даже для объектов с изогнутой поверхностью и независимо от угла их столкновения. Расчеты начинаются с учетом исходного расположения объектов, что позволяет быстро получать последовательность кадров, т.е. перемещение объектов.

Для избавления от мучений сотрудниками МП «МикС» создан пакет редактора спрайтов EOS, который возложил на компьютер всю неблагодарную работу по созданию программы рисования и блока данных спрайта. Редактор EOS автоматически настраивается на текущую операционную систему и может работать, а значит, создавать программу рисования как на ОС типа МикроДОС и CP/M, так и на РМУ. При этом работа пользователя с редактором предельно проста, освоение занимает 15 минут. Применение редактора в процессе обучения показало устойчивый интерес учащихся к предмету.

Новый способ моделирования найдет применение при изучении механики тел, где он позволяет составлять более сложные, чем в современных учебниках, примеры и задачи на трение. При этом, кроме обычных параметров (координаты и масса), можно задавать и такие, как трение при контакте объектов, и отображать перемещение объектов при любом ракурсе.

В Миннесотском университете разработан машинный метод построения графических изображений в гиперболическом пространстве, которое существенно отличается от традиционного евклидова пространства, например, неточным равенством 180° суммы всех углов в треугольнике или изменением расстояния между параллельными линиями и кривыми.

Математики Принстонского университета (шт. Нью-Джерси) используют этот метод при классификации трехмерных поверхностей по нескольким признакам в гиперболическом пространстве. По их мнению, исследование размещения этих поверхностей в гиперболическом пространстве позволит доказать, что все возможные многообразия определенного типа распределяются по строго ограниченному числу категорий. Кроме того, изучение многообразий позволяет точно

определять смысл уравнений, используемых для описания и моделирования физических явлений, а изображения в гиперболическом пространстве - более доходчиво пояснять и доказывать сложные математические постулаты.

Клиффорд Пиквер из Научно-исследовательского центра им. Т. Уатсона фирмы IBM (Йорктаун-Хайтс, шт. Нью-Йорк), автор многочисленных изображений, полученных методом компьютерной графики, создал свои наиболее впечатляющие изображения на основе теории хаоса. Эта теория изучает явления, для которых характерна зависимость от исходных условий. Хотя, отмечает Пиквер, на первый взгляд хаос кажется непредсказуемым, однако он подчиняется строгим математическим законам, которые можно сформулировать в виде уравнений.

Исследователи теории хаоса изучают дифференциальные уравнения, описыва-

ющие изменения во времени или в пространстве. При этом используются различные графические методы для отображения траекторий движения из разных исходных точек, особенно для уравнений, определяющих нерегулярное движение. На экране видеоиндикатора изображение строится с помощью частично накладывающихся друг на друга сфер, которые в результате образуют криволинейные поверхности, описываемые математическими уравнениями, а раскраска поверхностей придает им причудливый вид.

Таким путем удалось получить на экране самые различные и необычные изображения, в том числе морских ракушек, беззубой пасти, жирных червячков, ярких крыльев бабочек и т. п.

Science News, 24.1990, v.138, № 21, p.328.



Только Мы

**МОЖЕМ ПРЕДЛОЖИТЬ ВАМ
НАДЕЖНОГО ПОМОЩНИКА
В ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА!**

Учебно-игровой комплекс УИК-1,5 «Поиск» производства НПО «ЭЛЕКТРОНМАШ» — это современная недорогая система, состоящая из рабочего места преподавателя на базе компьютера «ПОИСК-2» и от 1 до 15 рабочих мест учащихся на базе микро-ЭВМ «ПОИСК-1.04», объединенных между собой локальной вычислительной сетью, размещенных на специальных регулируемых столах, программно совместимых с IBM PC.

НАШ АДРЕС:

252180, Киев-180, ул. Большая Окружная, 4.
НПО «Электронмаш».

ТЕЛЕФОНЫ:

476-26-35 отдел разработки
475-43-00 отдел маркетинга
474-12-34 отдел рекламы

Уважаемые господа!

Фирма ЛИНТех предлагает

ПОСЛЕДНИЕ НОВИНКИ ДЛЯ УЧЕБНЫХ КОМПЬЮТЕРНЫХ КЛАССОВ!

■ КУВТ «Корвет» с IBM-совместимой головной машиной: «NET-CP/M В-2.0»

Новая версия известной системы для КУВТ «Корвет» с IBM-совместимой головной машиной является модификацией системы «NET-CP/M В-1.0», подробно описанной в «Инфо» №2, 1992.

Предлагаемая система отличается от предшествующей дополнительными возможностями для программирования, позволяющими создавать мощное программное обеспечение, максимально используя возможности головного IBM PC/AT. Встроенные сетевые адаптеры новой версии, выполненные на самом современном технологическом уровне, по надежности и качеству изготовления не уступают вычислительной технике ведущих зарубежных производителей. Установка встроенных контроллеров может быть выполнена как пользователями, так и нашими специалистами.

Цена системы для стандартного класса (12 РМУ и 1 РМП) — 550 долларов США, оплата в рублях, безналичный расчет по официальному курсу ЦБ России. Гарантия — 1 год.

■ КУВТ УКНЦ с IBM-совместимой головной машиной: «NET-RT11 В-1.0»

Эта новинка, впервые созданная для КУВТ УКНЦ, не имеет аналогов, позволяет владельцам классов этого типа полностью избавиться от всех «сетевых» проблем и работать на каждом РМУ со стандартной системой «RT-11» без сбоев и «зависаний», обеспечивая мгновенный доступ к файлам на винчестере головного IBM PC/AT.

Сетевые контроллеры «NET-RT11 В-1.0» разработаны на базе встроенных адаптеров популярной системы «NET-CP/M В-2*» и обеспечивают скорость передачи информации по сети 375 кбит/с. Установка системы в классе практически не требует переделки стандартной сети УКНЦ.

Цена системы для стандартного класса (12 РМУ и 1 РМП) — 625 долларов США, оплата в рублях, безналичный расчет по официальному курсу ЦБ России. Гарантия — 1 год.

■ КУВТ PC/AT

Широкий спектр классов на базе импортных компьютеров PC/AT/286/386 SX/380. Вы сможете выбрать оптимальную конфигурацию, исходя из своих требований и финансовых возможностей, и постепенно наращивать мощность КУВТ благодаря специальной системе модернизации (Upgrade), предлагаемой только у нас. Возможна покупка КУВТ, состоящих из любого количества компьютеров.

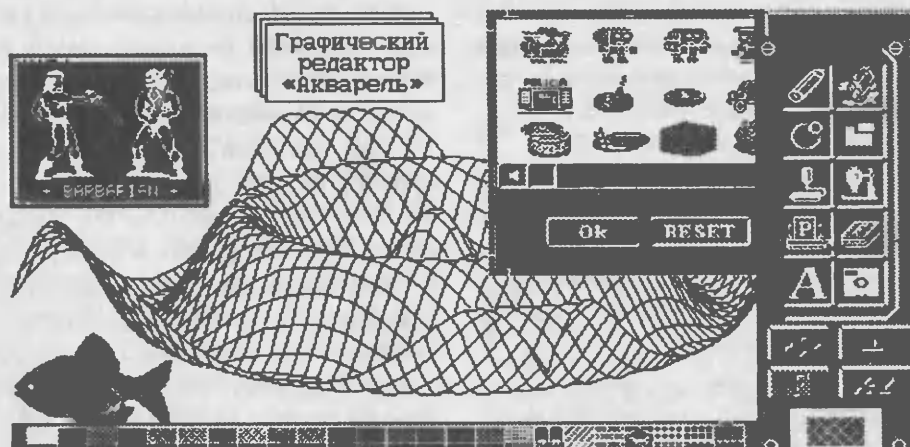
Цена на КУВТ PC/AT, состоящие из 11 РМУ и 1 РМП, — от 4000 до 14000 долларов США, в зависимости от выбранной конфигурации, оплата в рублях, безналичный расчет по официальному курсу ЦБ России. Гарантия — 1 год.



Адрес для корреспонденции:

119501, Москва, а/я 942.

Телефон: (095) 273-50-14.



В.В. Нужа

«ЛИНТех» ПРЕДСТАВЛЯЕТ ГРАФИЧЕСКИЙ РЕДАКТОР «АКВАРЕЛЬ»

Одной из важнейших характеристик любого современного персонального компьютера, вне всякого сомнения, являются его графические возможности. По этой части ПК8020 «Корвет» значительно превосходит подавляющее большинство своих восьмиразрядных собратьев: восемь основных цветов или градаций яркости при разрешении 512 на 256 точек и исключительно высоком быстродействии аппаратных графических функций позволяют использовать компьютер в самых разных графических приложениях.

Существовавшие ранее графические редакторы для этой машины обладали целым рядом недостатков. Как правило, они имели либо очень бедные функциональные возможности и низкий уровень сервиса, либо использовали оверлейные функции, непрерывно по мере необходимости загружаемые в оперативную память с диска. Последнее свойство крайне осложняет использование программ на рабочих местах учеников, так как по сети постоянно

пересылаются файлы большого объема, что делает невозможной одновременную работу таких программ на всех РМУ — большую часть времени они находятся в состоянии ожидания.

В 1992 году Лабораторией информационных технологий («ЛИНТех») был выпущен новый программный продукт — «Аквабель», свободный от перечисленных недостатков, и изначально ориентированный на работу в локальной сети. Написанный полностью на ассемблере, этот графический редактор сразу целиком загружается в ОЗУ и обращается к дискам лишь для считывания или записи создаваемых изображений, шрифтовых файлов и палитр. Цель настоящей публикации — ознакомить всех пользователей ПК8020 с основными возможностями этой программы.

Графический пакет «Аквабель», разработанный фирмой «ЛИНТех» для персональных компьютеров семейства «Корвет», включает в себя программы, предназначенные для выполнения графических

работ, а также демонстрационные и инструментальные файлы. Программы пакета выполнены в виде пиктограммного интерфейса и поддерживают работу с манипулятором мышь фирмы «Microsoft».

В базовую поставку пакета «Акварель» входят, кроме собственно графического редактора, редактор палитры, базовая и дополнительная палитра, палитра «штриховок», несколько популярных шрифтов (кириллица и латиница).

Графический редактор предназначен для создания и обработки изображений, содержащихся в графическом запоминающем устройстве компьютера, позволяет формировать прямоугольники, эллипсы, прямые и ломаные линии; рисовать пером и пульверизатором при помощи мыши или клавиатуры; производить различные перекраски участков экрана; копировать фрагменты экрана с одновременным зеркальным отображением, вращением и масштабированием; сохранять графический экран в виде твердой копии на принтере; работать с фрагментом экрана в системе масштабного редактирования; работать с заранее заготовленными матричными шрифтами и т. д.

Редактор предназначен для работы под управлением операционной системы CP/M версии 2.2, но может работать и с другими системами. Одной из привлекательных его особенностей является выход из программы с сохранением содержимого ГЗУ. Это достигается передачей управления из программы процессору консольных команд (ССР), для чего транзитная программа должна позаботиться о его сохранении. После того как программа загрузится, на экране появится рекламная заставка. Нажмите любую кнопку на клавиатуре, и вы окажетесь в среде пиктограммного интерфейса графредактора.

В качестве основного устройства для работы с редактором используется манипулятор «Microsoft mouse» или другие манипуляторы, которые могут эмулировать

«Microsoft», например известная мышь «Genius», либо стрелочное поле клавиш вашего компьютера, которое в сущности является клавишным имитатором мыши.

Для имитации перемещения мыши используются все стрелочные клавиши, а <0> и <cdot> имитируют соответственно левую и правую кнопки мыши.

Для перемещения курсора с помощью клавиатуры предусмотрены три основных режима — перемещение курсора на один пиксел, перемещение на заданное количество пикселов и перемещение на величину символа. Это позволяет с успехом пользоваться редактором при отсутствии мыши.

Экран редактора составляют рабочая область, в которой размещается редактируемое изображение, и панель управления, состоящая из набора пиктограмм, на которой нажатием кнопки (с помощью мыши или клавиатуры) производится выбор функций и модификация параметров. Панель занимает часть экрана, но содержимое ГЗУ под панелью не уничтожается. Размер рабочей области можно расширить во весь экран за счет временного выключения отображения панели.

Палитра располагается в нижней части экрана, ее составляют восемь базовых цветов компьютера и сорок составных. Набор составных цветов редактируется с помощью отдельной программы редактора палитры и может быть либо заранее инсталлирован с ее помощью, либо загружен в процессе работы с графредактором. В данной реализации элементом палитры служит прямоугольник размером 16x8.

Все функции редактора могут работать одновременно с двумя заранее выбранными цветами палитры, отображаемыми на цветоиндикаторе. Эти два цвета называются цветом фона и главным цветом. В качестве цвета фона может выступать только один из восьми базовых цветов (черный, синий, зеленый, голубой, красный, фиолетовый, желтый, белый), они занимают место в левом углу палитры. В

качестве главного цвета может выступать любой элемент палитры. Выбор цветов производится позиционированием курсора на нужную краску палитры и нажатием левой или правой кнопки мыши. При нажатии на левую кнопку мыши данный цвет палитры становится цветом фона, при нажатии на правую — главным цветом.

В правом нижнем углу панели находится цветоиндикатор, в котором присутствуют два цвета из набора палитры (главный цвет — в центре). С помощью цветоиндикатора можно произвести очистку всей рабочей области с закраской ее выбранными цветами. Для очистки экрана нужно установить курсор на пиктограмму цветоиндикатора и нажать одну из кнопок мыши. Нажатие левой кнопки мыши приводит к закраске экрана цветом фона, правой кнопкой экран закрашивается главным цветом.

В правом верхнем углу экрана находится индикатор положения курсора (ИПК). При нахождении курсора в рабочей области в нем отображаются текущие координаты курсора относительно текущего начала координат. Изначально начало координат находится в левом верхнем углу рабочей области, но может быть переустановлено. Выход из редактора осуществляется установкой курсора на ИПК и нажатием любой кнопки мыши. После этого в нижней части экрана появляется окно с надписью: «Сохранить изображение? (д, н)». При нажатии на клавишу «Д» производится выход из редактора с сохранением отредактированного изображения, при выборе «Н» производится выход с очисткой ГЗУ, при нажатии на любую другую клавишу выход отменяется и окно запроса исчезает.

С помощью модификатора пера можно выбрать одно из восьми перьев различной формы, которые используются при рисовании, построении линий, прямоугольников и эллипсов. Выбор пера производится установкой курсора на его пиктограмму и нажатием любой кнопки мыши. При этом

в окне производится последовательный перебор всех вариантов пера.

Размеры пера изменяются с помощью пиктограммы, содержащей цифры от 1 до 8. В функциях «Шрифт» и «Копировщик» пиктограмма задает увеличение размеров объектов в соответствующее число раз. В функциях «Ластик» и «Распылитель» — указывает размеры ластика и распылителя. В функции «Принтер» — указывает горизонтальную плотность печатаемого изображения. В функциях «Шрифт» и «Копировщик» можно растягивать изображение или литеру по заданным осям координат.

В графредакторе реализовано 14 базовых графических функций, которым соответствуют кнопки на панели редактора. Так как на панели изображено всего 10 кнопок, то четыре верхние кнопки имеют по две функции. Переключение функций кнопок производится одновременным нажатием левой и правой клавиш мыши.

Для рисования прямоугольника необходимо выбрать соответствующий режим, поместить курсор в рабочее поле и нажать любую кнопку мыши, при этом на экране появится надпись "Выберите вторую точку".левой кнопкой мыши производится выбор цвета фона, правой — главный цвет.

Теперь при перемещении курсора будет изображаться «резиновый прямоугольник», диагональными вершинами которого являются начальная точка и текущее положение курсора. Если переместить курсор в нужную вам позицию и нажать одну из кнопок мыши, то при этом происходит подтверждение или отмена процесса рисования прямоугольника. При нажатии на правую кнопку мыши производится рисование прямоугольника заданным цветом, на левую — отмена рисования. При отмене с рабочего поля исчезает «резиновый прямоугольник» и на экране появляется надпись «Отмена», которая сопровождается звуковым сигналом.

При рисовании линий и эллипсов все действия такие же, как и при построении прямоугольника.

С помощью функции «Ластик» можно производить замену одного цвета или группы цветов на новый цвет. Функция использует только число-модификатор, с помощью которого можно регулировать размер прямоугольной площади «ластика». Для перекраски следует установить курсор на нужный фрагмент экрана и нажать одну из кнопок мыши. При нажатии на левую кнопку мыши производится перекраска цвета фона на главный цвет. При нажатии на правую кнопку мыши производится перекраска всех цветов, не совпадающих с цветом фона, на главный цвет. Перекраска производится только внутри области, ограниченной размерами прямоугольного курсора «ластика», и невозможно по его периметру.

Функция «Распылитель» позволяет случайным способом распылять выбранную краску. Эта функция также использует только число-модификатор, с помощью которого можно регулировать размер прямоугольной области «распылителя». Распыление краски производится установкой курсора на нужный фрагмент экрана и нажатием одной из кнопок мыши. При нажатии на левую кнопку мыши производится закрашка цветом фона, на правую кнопку — главным цветом. Функция использует генератор псевдослучайных чисел «feedback shift register».

Функция «Закраска» позволяет заполнять область экрана заданным цветом. Функция аналогична оператору «PAINT» языка программирования BASIC. Чтобы произвести закрашку, нужно установить курсор в нужный вам участок экрана и нажать одну из кнопок мыши. Левая кнопка производит закрашку области главным цветом до границы, обозначенной цветом фона. Правая кнопка перекрашивает область, закрашенную цветом фона в главный цвет.

Функция «Закраска» — одна из самых сложных в редакторе. Для ее выполнения максимально используются ресурсы ОЗУ компьютера. Закраску можно производить как при включенной панели, так и при отключенной. Во втором случае для буферов выделяется больше памяти, так как используется область, которую обычно занимает содержимое ГЗУ под панелью. Поэтому при закрашке особо сложных областей рекомендуется отключать панель во избежание переполнения стека. При переполнении стека редактор выдает сообщение об ошибке и звуковой сигнал, сбой в работе программы не происходит.

Операцию закрашки области можно прервать с помощью клавиши <ESC>. Операция моментально прерывается, и выдается сообщение «Операция прервана», которое сопровождается звуковым сигналом.

С помощью функции «Масштабный редактор» можно редактировать отдельные пиксели фрагмента экрана. Существует два режима редактирования — одинарной или двойной плотности. Одинарная плотность позволяет редактировать фрагмент размером 32x24 пиксела. Пиксельная точка представляется квадратом размера 8x8. Двойная плотность позволяет редактировать фрагмент размером 64x48 пиксела, пиксельная точка также представляется квадратом 8x8.

С помощью функции «Принтер» можно сделать твердую копию содержимого экрана на принтере. Редактор «Акварель» работает с EPSON-совместимыми принтерами (EPSON FX-800, RAVI-800 и т.п.). При этом имеется возможность регулировать горизонтальную плотность строки на принтере.

Функция «Копировщик» позволяет копировать фрагменты экрана с одновременным масштабным увеличением изображения, зеркальным отображением и поворотом на 90, 180, 270 градусов. Операция копирования осуществляется методом «Экран-Буфер-Экран», что, соответ-

ственно, накладывает некоторые ограничения на размер области-источника, но зато позволяет неоднократно копировать заданный фрагмент экрана. Для выполнения этой функции необходимо, выбрав соответствующую пиктограмму, поместить курсор в рабочее поле и нажать любую кнопку мыши, при этом на экране появится надпись: «Выберите вторую точку». Теперь, при перемещении курсора, будет изображаться «резиновый прямоугольник». Далее перемещаем курсор в выбранную позицию и нажимаем на одну из кнопок мыши. Нажатием на правую кнопку мыши подтверждаем задание области-источника, на левую — отменяем установку.

После установки области-источника на экране появится прямоугольник области-приемника, который вы можете переместить в нужную вам позицию. На этом этапе можно повторно изменить форму области-приемника путем корректировки масштаба и осей масштабирования. Если прямоугольник на экране отсутствует — это означает, что его размеры превосходят размеры экрана, в таком случае нужно изменить модификаторы.

Копирование осуществляется после установки прямоугольника в нужную позицию, с последующим нажатием одной из кнопок мыши. При нажатии на правую кнопку производится полное цветовое копирование области-приемника. При нажатии на левую — осуществляется копирование, при котором цвет фона не выводится на экран. С помощью одновременного нажатия двух кнопок мыши производится отмена копирования.

Функция «Шрифт» предназначена для вывода на экран различных матричных символов, определенных в файле шрифта. При покупке пакета пользователю предлагается несколько стандартных шрифтов, которые могут помочь при разработке дизайна программ. Планируется выпуск отдельного пакета, в который бу-

дут входить разнообразные шрифты, и официальные владельцы «Акварели» будут письменно уведомляться при пополнении данного пакета.

В редакторе имеется возможность авто-смещения позиции при печати символа. При включенном автосмещении в ИПС вместо текущих координат отображается состояние клавиатуры, т. е. алфавит и регистр.

Программы пакета «Акварель» обладают повышенной надежностью при работе с диском и могут контролировать физические ошибки, при которых многие программы, написанные под CP/M-80, «вываливаются» в операционную систему и фактически уничтожают ваш труд.

«Акварельные» программы обладают богатой диагностикой ошибок обмена с диском, синтаксическим контролем имен файлов и автоматическим проставлением нужных расширений по умолчанию.

Для редактирования и создания палитр служит специальная программа, входящая в базовую поставку пакета. С ее помощью можно редактировать по пикселям палитровый набор, состоящий из сорока элементов. Элемент палитры представляет собой прямоугольную матрицу 16x8 пикселей, где каждый пиксел может принимать окраску одного из восьми базовых цветов «Корвета». Редактирование осуществляется аналогично работе с графическим редактором, с использованием мыши или дополнительного поля клавиатуры.

И, наконец, одно из важнейших свойств программы — простота использования. Для работы с графическим редактором «Акварель» не требуются специальные знания и навыки работы с персональным компьютером. Рисовать могут даже дети дошкольного возраста.

Приобрести «Акварель» и получить необходимые справки можно в фирме «Лин-Тех», рекламная информация которой приведена в этом журнале.

А. Никитин

LUT или «ячейка FAFB»?

В заметке А. Русских «Что может ячейка FAFB» («ИНФО», 1992/2, с. 67) автор сообщает о своем открытии: оказывается, управлять физическим цветом в Бейсике можно, минуя встроенную команду LUT! Все эффекты при этом связываются с некоей (вероятно, «волшебной») ячейкой FAFB.

Можно только приветствовать самостоятельный поиск, тем более что предлагаемые автором фрагменты программ в принципе работоспособны. И все же находки настолько слабы, а выводы — несостоятельны, что работа «не тянет» даже как доклад на школьном кружке. Главные промахи автора статьи — во второй ее части, где утверждает, что в Бейсике прямое программирование просмотрной таблицы (а ячейка FAFB является портом именно этого устройства) якобы эффективнее, чем стандартный бейсиковский LUT. И по объему памяти, и по времени, и по качеству картинки РОКЕ уступает LUT! Должен заметить — к счастью, уступает, так как обычно именно неэффективность «законных» средств Бейсика толкает на «преступления». В данном же случае такие мотивы могут возникнуть только по неопытности. Главный тезис, который предстоит сейчас подтвердить, заключается в следующем: никогда не следует без крайней нужды выходить за рамки разрешенных системой средств, так же как и использовать их неоговоренным образом. Кстати, это одна из основных идей, выдвигаемых таким авторитетным программистом, каким является Питер Нортон.

Почему программы А. Русских более расточительны по времени, чем LUT? Он даже привел два фрагмента параллельно, и на первый взгляд очевидно, что короткая программа автора заметки, имеющая лишь один цикл, эффективней более длинной, да еще и с тремя экранами, использующей LUT.

Но «на второй взгляд» оказывается, что все циклы LUT-программы инициализационные (т. е. однократно выполняются в самом начале про-

граммы), а авторский цикл работает каждый раз во время перезаписи порта. Конечно, команда LUT вызывает подпрограмму, где тоже есть подобный цикл, но это цикл в кодах микропроцессора, что на два-три порядка эффективнее, чем решение средствами Бейсика. Наконец, LUT работает «с оглядкой» на ход электронного луча кинескопа, что гарантирует высокое качество изображения (это подчеркнуто в примечании редакции к статье А. Русских).

Из сказанного очевидно, что в Бейсике лучше пользоваться командой LUT. Но можно и дальше бороться за эффективность программы, теперь уже в плане экономии памяти.

Вот одна из таких идей. Обратите внимание на то, что оператор LUT может адресоваться к любому элементу массива. Это позволяет в одной таблице «зарядить» множество цветовых эффектов. Конечно, массив будет больше 16 ячеек, но это, как правило, меньше чем несколько частично повторяющихся таблиц, заготовленных для каждого эффекта в отдельности. Чем лучше будет продумано устройство массива, тем меньше будет использовано памяти и тем удобнее (нагляднее, безошибочнее) будет программа в целом. Программисту останется лишь в нужный момент вызвать LUT с начальным элементом нужного фрагмента. Например, так:

```
10 DIM L%(39) ' Расход всего 80 байт
20 FOR Z=0 TO 7
30 L%(Z)-Z: L%(Z+24)-Z: L%(Z+32)-15
40 NEXT Z
```

Заметьте, что можно обойтись одним циклом там, где А. Русских применяет три. Этот фрагмент выполняется один раз в самом начале программы и не влияет на ее критические этапы. Теперь можно создавать множество эффектов одной лишь командой LUT:

520	LUT L%(0)	' Символы погашены, графика видна
...		
650	LUT L%(24)	' Мгновенно зажглись символы
820	LUT L%(8)	' Все погашено (и графика, и символы)
...		' Что-то рисуется, пишется...
950	FOR Z=0 TO 14	' Начало цикла постепенного (плавного)
960	LUT L%(9+Z)	' включения экрана
970	FOR I=0 TO 20:NEXT	' Задержка
980	NEXT Z	' Конец цикла

По примеру последнего цикла можно придумать много интересных эффектов. Наши кружковцы исследовали возможности LUT-мультипликации на примере вращающихся объектов (колес, пропеллеров и т. п.). Попробуйте, это короткие (10-15 команд), но очень занима-

тельные эксперименты. К тому же у вас появится объективный критерий для сравнения скорости и качества альтернативных приемов перепрограммирования цвета: чья вертушка быстрее крутится, у кого меньше помех на экране — тот и победил!

Д.Ю. Усенков

РЕАЛИЗАЦИЯ МНОГОЦВЕТНОЙ ЗАКРАСКИ НА БЕЙСИКЕ (по следам одной старой ошибки)

В журнале «Информатика и образование» №2 за 1989 г. была опубликована программа для реализации 16-цветной закрашки с помощью оператора Бейсика PAINT (автор – С.Зильберштейн). Кратко напомним, о чем шел разговор. Предлагалось с помощью опубликованной программы в кодах модифицировать Бейсик таким образом, чтобы иметь возможность закрашивать замкнутые области (с помощью PAINT) произвольно выбранным цветом из 16 возможных (из них четыре стандартных), где добавочные цвета задаются пользователем с помощью оператора описания DEF C<номер цвета>=(xxxx, xxxx, xxxx, xxxx), где "xxxx" – числа, кодирующие цвет.

К сожалению, в опубликованной программе оказались ошибки. Их поиск и устранение сильно усложнились из-за того, что коды были приведены в десятичном формате и отсутствовала контрольная сумма.

Сегодня среди пользователей БК уже имеется аналогичная программа в составе комплекта FBASIC, но она менее удобна – цвет приходится задавать байтами с помощью операторов POKE перед каждым вызовом PAINT, что более сложно и удлиняет программу.

Все вышесказанное побудило произвести подробный анализ, исправление

ошибок и доработку старого варианта программы. И теперь вашему вниманию предлагается программа, позволяющая выполнять все, что было обещано в упомянутом втором номере «Информатики и образования».

Ниже приводится листинг в кодах (с указанием контрольной суммы) и исходный текст на ассемблере в формате M18. Ассемблерный листинг не только поможет снизить вероятность ошибок при наборе на БК, но и даст возможность при желании легко перекомпилировать программу с другого начального адреса, задать другое количество цветов в палитре и т. д. Кроме того, весьма полезно рассмотреть ассемблер-текст поподробнее «в обучающих целях», так как он является хорошей иллюстрацией некоторых приемов работы Бейсик-транслятора и способов вмешательства в процесс компиляции.

«Постановка задачи программисту» или «немного о возможностях программы»

Прежде чем писать программу, программист четко формулирует для себя «техническое задание» – определяет все необходимые операции, которые должна обеспечивать программа. В нашем случае эта задача была поставлена в журнале «Информа-

тика и образование» №2 за 1989 г., но не решена.

Итак, наша программа должна обеспечивать:

1. Заливку с помощью PAINT неким дополнительным цветом (грамотнее было бы говорить о цвете и фактуре закрашки), заданным пользователем. При этом пользователь задает вид элементарного «макро-пиксела», состоящего из 4·4 точек.

2. Возможность указания номера выбранного для закрашки дополнительного цвета непосредственно в операторе PAINT, как обычно указывается стандартный цвет заливки. При этом программа должна обеспечивать распознавание номеров стандартных цветов (от 0 до 4) и нестандартных (от 5 до 15). Если задан стандартный цвет, заливка должна производиться как обычно.

3. Возможность задания цвета и фактуры для любого нестандартного «цвета» палитры с помощью оператора DEF C<номер>=(xxxx, xxxx, xxxx, xxxx), где «xxxx» — числа из 4 цифр, определяющие четыре строки «макро-пиксела»:



(цифры 1...4 обозначают общепринятые номера цветов точек, т.е. 1-красный, 2-зеленый, 3-синий, 4-черный)

Исходя из этого видно, что алгоритм нашей программы должен включать в себя как минимум три части: алгоритм закрашки, алгоритм перехвата и обработки ошибки при задании нестандартного номера цвета

в PAINT и алгоритм перехвата синтаксической ошибки при трансляции не предусмотренного в Бейсике оператора DEF C...

Отдельно рассмотрим одну из этих частей — алгоритм многоцветной закрашки. Остальные части достаточно хорошо просматриваются при анализе листинга программы на Ассемблере.

Алгоритм многоцветной закрашки

Проще всего получить новый цвет, загружая в служебные ячейки Монитора (@#212 — фон, @#214 — передний план) вместо стандартных байтов, кодирующих цвета (177777 — красный, 125252 — зеленый, 52525 — синий, 0 — черный), свои значения, кодирующие нужный цвет. Подробно об этом было рассказано в журнале «Вычислительная техника и ее применение» № 12 за 1990 г., с. 31. Однако простая подстановка имеет существенный недостаток: получаемый при заливке цвет оказывается всегда «в вертикальную полоску».

В программе GRAF3 применен более совершенный алгоритм: для первой телевизионной строки используется «основной» байт, для второй — тот же байт предварительно проворачивается влево и только затем заносится в ячейку @#214, для третьей строки — еще раз влево и т. д. Но и у этого алгоритма свой недостаток: он хорошо реализует простые «смесевые» цвета (темно-красный, темно-зеленый, темно-синий, желтый, фиолетовый и голубой, то есть состоящие из «сетки» точек двух каких-нибудь стандартных цветов), а также «гладкие» стандартные цвета. Но вот попытка задать «белый» цвет (смесь трех цветов) или задать более редкое расположение точек одного цвета по отношению к другому (например, на три черные точки одна зеленая) оканчивается неудачей: проступает явная «полосатость» получа-

емой закраски либо производится простая «заштриховка» наклонными линиями.

Мы же применим алгоритм еще более совершенный, и в то же самое время простой. Пусть пользователь сам задает четыре отдельных байта для закраски. Тогда можно будет при желании сделать и «сетку», и «штриховку», и сложную фактуру. А наша программа будет всего лишь подставлять в ячейку @#214 один из этих байтов в соответствии с координатой Y требуемой точки, то есть для нулевой строки – первый байт из четырех, для первой строки – второй, для второй – третий, для третьей – четвертый, для четвертой – снова первый и т. д.

Итак, новая программа должна перехватывать EMT30 и EMT32, определять, относятся ли эти функции к реализации оператора PAINT (ведь нам не нужно менять цвета для LINE, PSET и т. д.), вычислять по значению хранящейся в R2 координаты Y адрес байта в палитре, где записаны по 4 байта на каждый цвет, заносить выбранный байт в ячейку @#214 и после установки точки или линии не забывать восстановить старый цвет рисования. Проще всего это реализовать так: сохраним в стеке значение R2, выделим из него три младших бита – останется число от 0 до 3, прибавим это число к хранящемуся в ячейке @#300 (она используется только при операциях обмена с магнитофоном и мы можем использовать ее как буфер) адресу первого байта палитры, соответствующего указанному ранее номеру цвета, занесем байт в ячейку @#214, сохранив предварительно в стеке старое значение из этой ячейки, восстановим истинное значение R2 и обработаем EMT30 или EMT32, а затем восстановим из стека старое значение ячейки цвета.

Именно так и сделано в программе. Мы можем увидеть это в первой части листинга, «посвященной» EMT-перехвату.

Программа начинается с EMT-драйвера. Мы перехватываем на него EMT-прерывание и по сохраненному в стеке адресу возврата определяем точное место, откуда был вызван EMT (из PAINTа, из LINE или откуда-нибудь еще). Для такой работы полезно иметь ассемблеровский листинг Бейсик-транслятора. (Этот листинг, а также листинг Монитора БК-0010 редакция планирует опубликовать в одном из выпусков приложения к ИНФО – «Персональный компьютер БК-0010 – БК-0011М».)

Если «пойманный» EMT не тот, который нам нужен, производим переход на стандартный драйвер обработки EMTов по адресу 100112.

Кстати, нет нужды делать отдельные реализации замены цвета для EMT30 и EMT32. В нашей программе производится определение (опять-таки по адресу, с которого был вызван EMT) типа функции (EMT30 или EMT32) и нужный машинный код заносится непосредственно «в тело» самой программы, а затем, после замены цвета, выполняется.

Далее, нам нужно производить перехват ошибок Бейсика. Для этого мы пишем свой драйвер TRAP-прерываний. И первая же строка нашей программы, первое же действие, производимое при любом EMT-прерывании – занесение нового значения адреса обработки прерываний TRAP. Ну а в самом драйвере TRAP мы производим аналогичным способом (по адресам вызова) анализ TRAP-функций, чтобы выделить среди всех возможных вызовов сообщений об ошибках, а также (не следует об этом забывать) вызовов TRAP-подпрограмм самого Бейсика нужные нам ошибку цвета в PAINT и ошибку синтаксиса в DEF. Остальные TRAPы перенаправляем на стандартную обработку.

Если посмотрим на ассемблеровский листинг, то увидим три анализируемых случая TRAP. Во-первых, мы перехватываем первый же служебный TRAP при компиляции оператора PAINT. Это не ошибка, а признак «для нас»: пора вставить в шитый код адрес маленькой самодельной подпрограммки, которая при работе уже откомпилированной Бейсик-программы будет производить исходное обнуление буферной ячейки @#300 перед каждым оператором PAINT (зачем — обсудим позже).

Во-вторых, отлавливается сообщение об ошибке, когда при работе откомпилированной Бейсик-программы БК «замечает», что в PAINT задан номер цвета более 4. Теперь нам нужно «сделать вид», что ошибки не было. Мы прежде всего определяем «оставшийся» в R0 после компиляции заданный нами нестандартный номер цвета и вычисляем по нему адрес первого байта из четырех отведенных для этого цвета в палитре. Записываем адрес в ячейку @#300. Заменяем «условно» цвет закраски на текущий (0), чтобы Бейсик «больше не ругался», и «как ни в чем не бывало» продолжаем работу, переходя к следующему шитому коду. (Разумеется, уже произведена «нормализация» стека, то есть с помощью команды CMP (R6)+, (R6)+ удалены занесенные туда при вызове TRAP четыре байта.)

Затем, уже при постановке точек и линий, EMT-перехватчик заменит «условный» цвет заливки на заданный нами через значение ячейки @#300. Заметим, что если мы задали в PAINT стандартный номер цвета от 0 до 4, то этой ошибки не возникнет, ячейка @#300 останется обнуленной, как это сделала подпрограммка, рассмотренная нами «во-первых», а при отработке EMT наша программа не будет заменять цвет закраски, а просто выведет точку или линию заданным стан-

дартным цветом. Таким образом реализовано одно из дополнительных условий задачи — если цвет стандартный, закраска производится как обычно.

И в-третьих, отлавливаем ошибку при компиляции DEF. Точно так же нормализуем стек, а затем выполняем вместо вильнюсского транслятора его работу, то есть сами компилируем наш оператор DEF C<n>=... .

Для этого вначале убираем лишние пробелы между «DEF» и «C» с помощью служебной функции TRAP112. Проверяем, совпадает ли текущий символ исходного текста на Бейсике с символом «C» (латинское заглавное). При компиляции регистр R3 содержит адрес текущего байта исходного текста и мы это используем. Если текущий символ не «C», значит, мы действительно ошиблись в Бейсик-программе, и мы генерируем запоздалое TRAP2. Если не ошиблись, продолжаем. Снова убираем незначачие пробелы (если они есть) между «C» и номером цвета с помощью TRAP112. Вызываем подпрограмму, считающую записанное в десятичной форме в исходном тексте число — номер цвета в палитре и заносим номер в R5. Подпрограмма чтения номера (NUM) сама следит и за синтаксисом (чтобы номер не содержал нецифровых символов), и за значением номера (не менее 5 и не более 15), выдавая в некорректном случае «ОШИБКУ 2». Вновь вызываем TRAP112 для ликвидации пробелов перед знаком равенства и проверяем, действительно ли это знак равенства.

Теперь с помощью другой служебной функции TRAP127 производим поиск открывающей скобки (TRAP127 пропускает пробелы в исходном тексте и возвращает в R3 адрес первого непробельного байта после найденной открывающей скобки). И с помощью подпрограммы PPR анализируем четыре записанных через запятую чис-

ла. Сначала первое, мы «на нем стоим». Потом вызываем спецфункцию TRAP126, которая выводит нас аналогично TRAP127 на первый непробельный байт после запятой, и снова вызываем подпрограмму PPR для второго числа. Сделав так четыре раза для четырех чисел, вызываем TRAP130, которая, аналогично двум предыдущим спецфункциям, возвращает адрес после закрывающей скобки и, наконец, с помощью TRAP106, отсекающей попутно комментарии, переходим на следующую строку исходного Бейсик-текста. И вот теперь возвращаемся в стандартный компилятор.

Как работает подпрограмма PPR, думаю, не стоит разбирать подробно. Это можно сделать и самостоятельно по ассемблер-тексту.

И в качестве своеобразного «резюме» повторим еще раз сведения об используемых при компиляции регистрах процессора и служебных TRAP-подпрограммах.

ПРИ КОМПИЛЯЦИИ ИСПОЛЬЗУЮТСЯ:

1. Регистры:

R3 — адрес текущего байта исходного текста Бейсик-программы (указатель на текущий байт);

R1 — адрес текущей ячейки для записи шитого кода.

2. Служебные TRAP-функции:

TRAP112 — тестирует текущий байт исходного текста и, если это пробел, смещает указатель «вправо», пока не встретит непробельный символ. Таким образом, при компиляции реализуется пропуск незначущих пробелов;

TRAP127 — аналогичным образом пропускает пробелы, а затем проверяет: если очередной, непробельный символ — открывающая скобка («(»), снова пропускает пробелы после скобки и возвращает в R3 адрес первого непробельного символа в

скобках. Если не скобка, генерируется ОШИБКА 2;

TRAP126 — аналог предыдущей функции, «натасканный» на запятую («,»);

TRAP130 — аналог двух предыдущих по отношению к закрывающей скобке («)»);

TRAP106 — пропускает все байты исходного текста, пока не встретится код 12 — код конца строки. Возвращает в R3 адрес первого байта следующей строки и используется как «завершающий аккорд» при компиляции каждой из строк Бейсик-программы. Кстати, отсекает дополнительные комментарии;

TRAP с номерами менее 100 генерируют текст «ОШИБКА п», где п — номер ошибки, равный номеру TRAP-функции. В Бейсик-трансляторе задействованы не все номера ошибок из возможных, так что при желании вы можете предусмотреть в своих программах свои номера сообщений об ошибках. Нужно только подать TRAP<номер>, а остальное Бейсик сделает сам.

Правила работы с программой

Программа в машинных кодах 16PAIN.BIN должна быть считана в Бейсик с адреса &O37300 с помощью оператора BLOAD. Предварительно нужно выполнить оператор CLEAR ,&O37300. Перед компиляцией (то есть ДО ввода команды RUN!) нужно занести адрес начала программы (в нашем случае 37300) в ячейку @#30 оператором POKE &O30,&O <нач. адрес>, набранным в непосредственном режиме. После этого Бейсик-программа запускается на выполнение как обычно.

В Бейсик-программе допустимы операторы:

DEF C <номер> = (xxxx, xxxx, xxxx, xxxx) — задание цвета, <номер> — от 5 до 15;

PAINT (X,Y), <номер>, <цвет границы> — закраска внутри контура, <номер> — от 0 до 15, <цвет границы> — от 0 до 4.

Желающие могут перекомпилировать ассемблерный текст в системе типа M18, заменив начальный адрес программы на требуемый (программа неперемещаемая). При этом можно также расширить размер палитры, для чего в помеченных звездоч-

ками строках нужно изменить значения констант:

#460 заменить на большее, увеличивая на 4 для каждого добавляемого цвета (при этом уменьшается «действующий» объем стека и вы тем самым уменьшаете возможности Бейсик-программы);

#17 заменить на новый максимально допустимый номер цвета в палитре.

; ;
; Драйвер многоцветной графики для оператора PAINT Вильнюсского
; Бейсика (исправленный вариант программы из «ИНФО» №2 за 1989г)

----- SCREW(c) -----

; ;
; Усенков Д.Ю. МОСКВА 1992
; ;

; ;
;%EMT: MOV #%TRAP,@#34

 CMP @R6,#126264

 BNE I1

I2: MOV #104030,KOMMD

 BR I3

I1: CMP @R6,#126310

 BEQ I2

 CMP @R6,#126320

 BNE A1

 MOV #104032,KOMMD

I3: MOV @#214,-(R6)

 TST @#300

 BEQ KOMMD

 MOV R2,-(R6)

 CMP R6,#460 ; *

 BNI I4

 MOV 2(R6),@#214

 TRAP 7

I4: BIC #177774,R2

 ADD @#300,R2

 MOVB @R2,@#214

 MOV (R6)+,R2

KOMMD: EMT 30

 MOV (R6)+,@#214

 RTI

A1: JMP @#100112

; Новый EMT-драйвер:

; Новый вектор TRAP

; Проверка: откуда вызван EMT?

; EMT30 реализации PAINT?

; ;

; ДА!

; ;

; EMT30 также из PAINT?

; ДА-переход, нет- проверка:

; EMT32 из реализации PAINT?

; ;

; Старый цвет закраски — в стек

; Цвет закраски-стандартный?

; ДА-сразу на вывод точки/линии

; НЕТ-формируем цвет закраски:

; Не затер ли стек палитру цветов?

; НЕТ-формируем цвет

; ДА-восстановим старый цвет за-

; краски и выводим "ОШИБКУ 7"

; Формируем цвет заливки по коор-

; динате Y и номеру цвета из ячейки

; @#300

; Восстанавливаем значение Y

; Рисуем точку/линию

; Восстанавливаем цвет закраски.

; Выход из EMT

; Для других EMT-на стандартный

		; драйвер
ERR:	JMP @#146404	; Стандартный драйвер TRAP
		; Новый драйвер TRAP:
%TRAP:	CMP 4(R6),#145230	; Компиляция PAINT?
	BNE A2	;
	MOV #X1V,(R1)+	; ДА-вносим в шитый код свой код
	BR ERR	; и продолжаем трансляцию
X1V:	CLR @#300	; Подпрограмма предварительного
	JMP @(R4)+	; обнуления номера цвета
A2:	CMP @R6,#125172	; Ошибка при обработке PAINT: не-
		; стандартный номер цвета?
	BNE A3	;
	CMP (R6)+,(R6)+	; ДА-нормализуем стек,
	SUB #5,R0	; вычисляем по номеру заданного в
	ASL R0	; PAINT цвета номер в палитре, по-
	ASL R0	; мещаем его в ячейку @#300, для
	ADD #400,R0	; "маскировки" ошибки назначаем
	MOV R0,@#300	; цвет, равный 0, и возвращаемся
	CLR @R6	; в программу обработки PAINT
	JMP @(R4)+	;
A3:	CMP @R6,#145566	; Ошибка при компиляции DEF Sp=..?
	BNE ERR	; НЕТ — на стандартный драйвер
	CMP (R6)+,(R6)+	; ДА-нормализуем стек,
	TRAP 112	; убираем лишние пробелы,
	CMPB (R3)+,'C'	; это наш DEF C?
	BNE SYERR	; Нет-ошибка 2
	TRAP 112	; убираем лишние пробелы после "C",
	JSR R7,NUM	; считываем номер цвета
	BR B	;
SYERR:	TRAP 2	; выход на сообщение об ошибке
B:	TRAP 112	; убираем лишние пробелы перед "="
	CMPB (R3)+,'='	; дальше идет код "="?
	BNE SYERR	; НЕТ-ошибка 2
	TRAP 127	; ищем открывающую скобку
	SUB #5,R5	; формируем адрес ячеек палитры
	ASL R5	; для данного цвета
	ASL R5	;
	ADD #400,R5	;
	MOV #3,-(R6)	; Формируем составляющие байты
A:	JSR R7,PPR	; п/пр чтения очередного числа
	TRAP 126	; ищем запятую
	DEC @R6	;
	BGT A	;
	TST (R6)+	;
	JSR R7,PPR	; последнее число
	TRAP 130	; поиск закрывающей скобки
	TRAP 106	; выход на следующую строку Бейсик

```

; программы
RTS R7 ; выход на продолжение компиляции
;
PPR: CLR -(R6) ; Подпрограмма формирования байтов:
MOV #4,R2 ;
AA: ASR @R6 ; загружает очередное 4-цифровое
ASR @R6 ; число из списка и формирует по
MOVB (R3)+,R4 ; нему байт для палитры
SUB #64,R4 ;
NEG R4 ;
CMP R4,#3 ;
BHI SYERR ;
SWAB R4 ;
ASR R4 ;
ASR R4 ;
BIS R4,@R6 ;
SOB R2,AA ;
MOV (R6)+,R4 ;
MOVB R4,(R5)+ ; Готовый байт заносим в палитру
RTS R7 ;
;
; Подпрограмма чтения цвета:
NUM: MOV R0,-(R6) ;
CLR R5 ;
1: CMPB @R3,'0' ; заносит в R5 номер цвета, за-
BLO 2 ; данного в описании DEF C<номер>
CMPB @R3,'9' ;
BHI 2 ;
ASL R5 ;
MOV R5,R0 ;
ASL R5 ;
ASL R5 ;
ADD R0,R5 ;
MOVB (R3)+,R0 ;
SUB '0',R0 ;
BIC #177400,R0 ;
ADD R0,R5 ;
BR 1 ;
2: CMP R5,#5 ;
BLO 3 ;
CMP R5,#17 ; * ; Ограничение кол-ва цветов
BHI 3 ;
MOV (R6)+,R0 ;
RTS R7 ;
;
3: TRAP 2 ; Ошибочный номер цвета
;
;-----

```

Кодовый листинг программы 16PAIN.BIN

НАЧ. АДРЕС : 037300

ДЛИНА : 000500

012737	037436	000034	021627	126264	001004	012767	104030
000074	000411	021627	126310	001771	021627	26320	001033
012767	104032	000050	013746	000214	005737	000300	001417
010246	020627	000460	101004	016637	000002	000214	104407
042702	177774	063702	000300	111237	000214	012602	104030
012637	000214	000002	000137	100112	000137	146404	026627
000004	145230	001006	012721	037454	000767	005037	000300
000134	021627	125172	001013	022626	162700	000005	006300
006300	062700	000400	010037	000300	005016	000134	021627
145566	001343	022626	104512	122327	000103	001004	104512
004767	000134	000401	104402	104512	122327	000075	001373
104527	162705	000005	006305	006305	062705	000400	012746
000003	004767	000022	104526	005316	003373	005726	004767
000006	104530	104506	000207	005046	012702	000004	006216
006216	112304	162704	000064	005404	020427	000003	101333
000304	006204	006204	050416	077216	012604	110425	000207
010046	005005	121327	000060	103417	121327	000071	101014
006305	010500	006305	006305	060005	112300	162700	000060
042700	177400	060005	000756	020527	000005	103405	020527
000017	101002	012600	000207	104402	000000	000000	000000

КОНТРОЛЬНАЯ СУММА : 011047

И.А. Сапегин

О ПРИРУЧЕНИИ КЛАВИШИ "СТОП"...

В ИНФО №2 за 1992 г. была напечатана статья Д.Якошвили "Обработка прерываний по клавише СТОП средствами Бейсика". Способ определения адреса начала "шитого кода" строки, предложенный автором, имеет несколько недостатков. Во-первых, он очень трудоемкий. Во-вторых, даже при небольшом изменении программы приходится проводить всю работу заново.

Я предлагаю программу, свободную от этих недостатков:

```

1 FOR I=PEEK(1044)+4 TO 16384
  STEP 6
2 IF ABS(65536*(SGN(PEEK(I))-
  1))/2+PEEK(I)= <A> THEN POKE
  514,PEEK(I-4) ELSE 7
3 POKE 512,5572
4 POKE 516,92
5 POKE 4,512
6 GOTO 10
7 NEXT I
10 REM ...ВАША ПРОГРАММА...
```

Во второй строке <A>—это номер строки в вашей программе, куда нужно перейти при нажатии клавиши СТОП.

При запуске программа сама определит адрес начала "шитого кода" нужной строки. Для этого используется таблица номеров строк, адрес начала которой записан в ячейке 1044 (здесь и далее все адреса десятичные). В этой таблице для каждой строки записано три параметра: адрес начала "шитого кода" строки, адрес начала ее исходного текста и номер.

Моя программа последовательно просматривает все номера строк в этой таблице и, найдя требуемый, записывает адрес начала "шитого кода", соответствующего нужной строке, в ячейку 514.

Подпрограмма в машинных кодах, расположенная в памяти с адреса 512 и заносимая туда строками 3 и 4, работает так, как написано в статье Д. Якошвили.

ОТ РЕДАКЦИИ

Письмо И.А. Сапегина продолжает начатую в прежних публикациях тему о "приручении" клавиши СТОП в Бейсике. Предлагаемая им программа позволяет "подвешивать" клавишу СТОП на нужную строку полностью автоматически, не производя при этом длительных и запутанных ручных расчетов и поисков, описанных в статье Д. Якошвили.

Хотелось бы добавить к сказанному автором следующее. Во-первых, в программе не предусмотрен контроль за существованием строк, и если в качестве <A> вы зададите номер

несуществующей строки, реакция БК будет, мягко говоря, непредсказуемой. Во-вторых, если ваша программа не изменяет искусственно при своей работе шитый код и содержимое ячейки @#4, то после выхода из программы в диалоговый режим по оператору STOP или END нажатие на клавишу СТОП по-прежнему вызывает переход на заданную Бейсик-строку. Таким образом данный эффект дает возможность повторного запуска Бейсик-программы, заменяя собой команду RUN <номер строки> или GOTO <номер строки>.

Р. Аскеров

ЕСЛИ НЕТ ПРИНТЕРА...

... и вам нужна копия с экрана БК, и не одна — как быть? Если у вас есть фотоаппарат и вы можете подключить свой компьютер к телевизору с размером экрана по диагонали 50-60 см, то нет проблем!

После длительных экспериментов в Брянском клубе пользователей БК разработана методика съемок с экрана ТВ. Использовался дальномерный фотоаппарат ФЭД5В и черно-белый телевизор КАСКАД-230 с диагональю 59 см. Аппарат устанавливается на штативе (впрочем, снимать можно и «с рук», жертвуя при этом четкостью) на расстоянии 1 м от экрана, видеоискатель наводится на точку, отстоя-

щую на 13 см от верхней кромки экрана и на 23 см от левой. На шкалах фотоаппарата устанавливаются значения выдержки 1/30, диафрагмы 2.8, дальности 1 м. Спуск затвора можно производить как через фототросик, так и непосредственным нажатием на кнопку.

На негативах изображение получается четким, во весь кадр, и в дальнейшем может быть увеличено в любое количество раз и переведено на фотобумагу. К сожалению, при использовании некоторых видов пленки изображение может оказаться слегка неравномерным по яркости из-за особенностей телевизионного экрана.

ПРИМЕЧАНИЕ РЕДАКТОРА

Данный совет полезен не только пользователям БК, но и всем, кто работает с домашними ПЭВМ, тем более что многие «самодельные» компьютеры вообще не рассчитаны на подключение принтера. Практическая ценность предложенного Брянским клубом пользователей БК совета значительно повышается и за счет того, что в нем приведены готовые рекомендации при первьсъемке изображения на экране с помощью фотоаппарата с видеоискателем, а работать таким фотоаппаратом в подобных случаях намного сложнее, чем зеркальной фотокамерой.

Следует также сказать, что и на профессиональных персональных компьютерах в составе графических станций, оснащенных принтерами и графопостроителями, также используется фотографическая пересъемка изображения на экране с помощью специализированных фотокамер. Подробнее об этом можно узнать из книги

«Компьютер обретает разум» (М.: Мир, 1990, с. 138).

Однако справедливости ради следует заметить, что предлагаемая идея использования фотоаппарата для получения копии экрана на БК не нова. О ней, в частности, говорилось на страницах журнала «Наука и жизнь» (1986, №10, с. 113, заметка читателя И. Львова из Москвы). В «Науке и жизни» предлагалось производить пересъемку с экрана зеркальной фотокамерой «Зенит» со штатива в затемненной комнате (для устранения бликов на экране). Объектив «Волна-9», пленка 32 или 65 ед ГОСТ. Диафрагма 5.6 — 8 с выдержкой 3 — 4 с для 32 ед или 1 — 2 с для 65 ед (выдержка отрабатывается вручную). Проявитель стандартный, время проявления — указанное на коробочке с пленкой.

*Распространением журнала «Информатика и образование»
в зарубежных странах занимается Акционерное общество
«Международная книга»*

*через своих конгр-агентов в соответствующих странах.
Адреса фирм-агентов А/О «Международная книга»*

Вы можете узнать

*в редакции журнала или в А/О «Международная книга»:
117040, Россия, Москва, улица Большая Якиманка, 39.*

Факс: (095) 238-46-34
Телефон: (095) 238-49-67
Телекс: 411160
Индекс журнала: 70423
Периодичность: 6 номеров в год

*Our journal is exported by Joint-Stock Company
«Mezhdunarodnaya Kniga»
through their agents around the world.
Address of the company «Mezhdunarodnaya Kniga»:
17049, Russia, Moscow, Bolshaya Yakimanka, 39.*

Telefax: (095) 238-46-34
Phone: (095) 238-49-67
Telex: 411160
Index: 70423
Issues per year: 6
**Price for year
included air delivery:** 39 US Dollars

М.Е. Степанов

Внутренний голос: Давай еще программу напишем.
 Программист: Я уже устал.
 Внутренний голос: Ну, ты как хочешь, а я напишу.
 (Старинный анекдот)

ЗВЕЗДОЧКИ, ДУГИ И ВНУТРЕННИЙ ГОЛОС

Разделим окружность на равные части p точками, а затем будем последовательно соединять отрезками очередную точку с m -й по счету от нее точкой. В итоге получим правильный звездчатый многоугольник, вписанный в окружность. Компьютер позволяет строить такие звездочки для достаточно больших значений p и m (рис. 1).

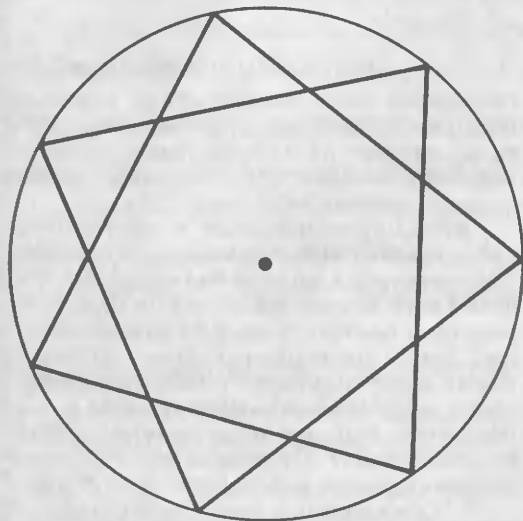


Рис. 1

```

10 COLOR 10,1,1:SCREEN 2
20 N=7:M=2:R=90:X0=128:Y0=96
30 PI=4*ATN(1):A2=PI/2
40 ' circle(x0,y0),r: ' МОЖНО УБРАТЬ
    ПЕРВЫЙ ПРИЗНАК КОММЕНТАРИЯ
50 GOSUB 1000: ' ПОДГОТОВКА
    ВСПОМОГАТЕЛЬНЫХ ВЕЛИЧИН
    
```

```

60 FOR I=1 TO N
70 A1=A2:A2=A2+DA:IF A2>2*PI THEN
    A2=A2-2*PI:A1=A1-2*PI
80 GOSUB 2000: ' ОЧЕРЕДНАЯ
    СТОРОНА ЗВЕЗДОЧКИ
90 NEXT I
100 GOTO 100
1000 ' ПОДГОТОВКА ВЕЛИЧИН
1010 DA=2*PI*M/N
1020 RETURN
2000 ' СТОРОНА ЗВЕЗДОЧКИ
2010 X1=X0+R*COS(A1)
2020 Y1=Y0-R*SIN(A1)
2030 X2=X0+R*COS(A2)
2040 Y2=Y0-R*SIN(A2)
2050 LINE (X1,Y1)-(X2,Y2)
2060 RETURN
    
```

Красиво! Но все же внутренний голос подсказывает, что прямые линии не уравниваются с окружностью. Если бы они прогнулись в сторону центра, звездочка стала бы воздушней, изящней, соразмерней (рис. 2).

Самые острые и колкие уголки у звездочек получаются, если дуги будут касаться радиуса исходной окружности. Если одна окружность касается двух радиусов другой

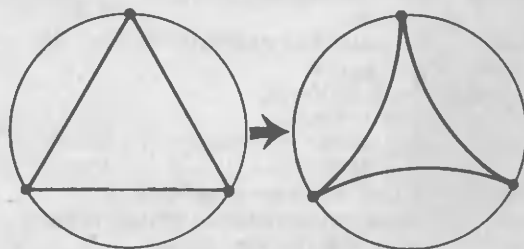
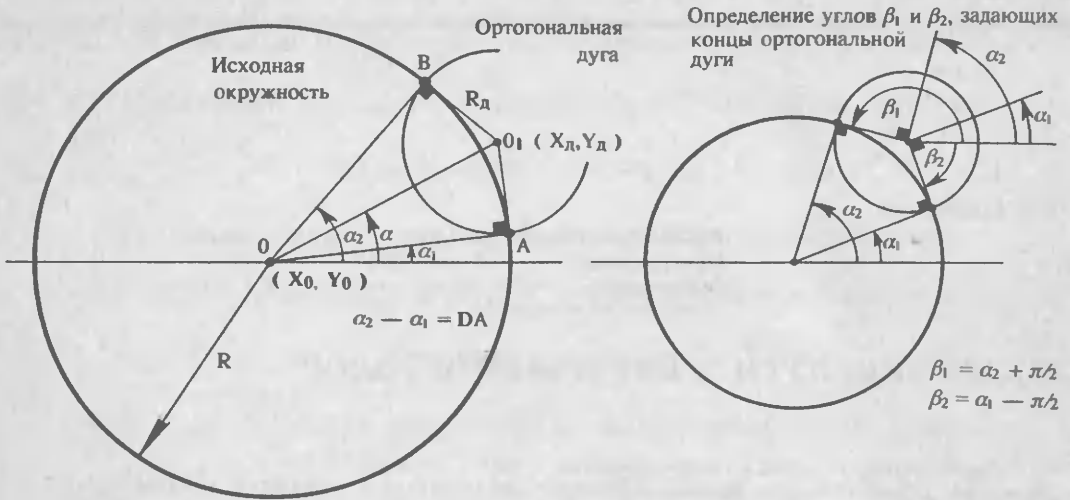


Рис. 2



Вывод формул:

$$\alpha = \alpha_1 + \frac{\alpha_2 - \alpha_1}{2} = \frac{\alpha_1 + \alpha_2}{2}$$

$$L = OO_1 = \frac{R}{\cos \frac{\alpha_2 - \alpha_1}{2}}$$

$$R_d = OA - OB = R \cdot \operatorname{tg} \frac{\alpha_2 - \alpha_1}{2}$$

$$X_d = X_0 + L \cdot \cos \alpha$$

$$Y_d = Y_0 - L \cdot \sin \alpha$$

Рис. 3

(в их концах), то окружности называются ортогональными. Мы собираемся строить звездочки из дуг, ортогональных к исходной окружности. Чтобы сделать это достаточно успешно, нужно получить формулы для нахождения по двум концевым точкам параметров дуги, ортогональной окружности (рис. 3).

Заданы: радиус исходной окружности R , координаты ее центра X_0, Y_0 и концы дуги AB , задаваемые углами α_1 и α_2 .

```

1000 ' ПОДГОТОВКА ВЕЛИЧИН
1010 DA=2*PI*M/N
1020 L=R/COS(DA/2)
1030 RD=R*TAN(DA/2)
1040 RETURN
2000 ' СТОРОНА ЗВЕЗДОЧКИ
2010 A=(A1+A2)/2
2020 XD=X0+L*COS(A)
2030 YD=Y0-L*SIN(A)
2040 B1=A2+PI/2
2050 B2=A1-PI/2
2060 IF B1>2*PI THEN B1=B1-2*PI
2070 IF B2<0 THEN B2=B2+2*PI:GOTO 2070
2080 CIRCLE(XD,YD),RD,,B1,B2
2090 RETURN

```

Следует помнить, что в случае, когда величина DA близка к π , радиус RD орто-

гональной дуги может стать слишком большим. При этом падает точность. Дуги то не доходят до нужной точки, то перескакивают через нее. Наконец, может произойти и переполнение.

Дуги, ортогональные к окружности, представляют собой прямые на плоскости Лобачевского в модели Пуанкаре. Обсуждать такой способ в деталях не будем, но советуем прочитать одну из многочисленных книг, посвященных этому вопросу. Перед вами откроется новый необычный мир и, несомненно, возникнут идеи новых программ. Еще раз прислушаемся к внутреннему голосу. Он продиктует несколько дополнительных заданий.

1. Объедините в одной программе дуги и отрезки.

2. Внутренние ортогональные дуги через одну замените на внешние. Вместо звездочки получится волна на сфере.

3. Нарисуйте пучок ортогональных дуг, выходящих из одной точки. При этом помните о тех случаях, когда DA близко к π .

4. Нарисуйте одновременно два или более таких пучков.

5. Прислушайтесь к своему внутреннему голосу. О том, что услышите, сообщите в редакцию журнала.

К КОЛЛЕДЖ

Фирма "КОЛЛЕДЖ":

Мы помогаем Вам в решении проблем компьютеризации учебных заведений



К Вашим услугам:

★ **Поставка компьютерных классов "под ключ"** на базе IBM – совместимой техники различной конфигурации, УКНЦ, БК – 0011М;

★ **Изготовление мебели** для учебных заведений (парты, столы, стулья, скамейки, стеллажи и др.)

Доставка техники и мебели осуществляется по Москве и Московской области.

Последние программные и методические разработки:

★ Программно – методический комплекс "Информатика" для УКНЦ, КУВТ – 86, ДВК, IBM.

★ Пакет административных программ для УКНЦ ("Завуч", "Мини – канцелярия", "Бухгалтерия", "Склад и реализация").

★ Информационно – поисковая система "DATA – MANAGER" (База данных, электронные таблицы, текстовый редактор в едином интерфейсе).

★ Обучающие программы по общеобразовательным предметам (физика, экономика, русский язык).

Наш адрес: 107005, Москва,
Волховский пер., д.11,
Фирма "Колледж"

Телефоны: 265-62-65
261-25-13

Телефакс: 265-62-65

Каталог программного обеспечения высылается бесплатно

Цена по подписке:
для индивидуальных подписчиков 120 руб. (индекс 70423).
для предприятий и организаций 200 руб. (индекс 73176).
В розницу цена договорная.
Информатика и образование. 1993. №4. 1-128.

75-602

