

ИНФОРМАТИКА И ОБРАЗОВАНИЕ

2 1993



ПРОГРАММНЫЕ ПРОДУКТЫ

для Вас!

TEACHCAD v3.5 - конструктор гиперсред. Не зная ни одного языка программирования, Вы сможете построить графическую базу данных, создать "живой" отчет, смоделировать какое-либо действие или лекцию по интересующей Вас теме. Создание нужной Вам программы перестанет быть проблемой и займет всего 10% Вашего времени. TeachCAD - это новая технология для Вас!

JUNIOR-1 v2.0 - набор развивающих игр для детей 7 - 12 лет, разбитый на три части разной сложности. Junior-1 это не просто программы! Это программно-методический комплекс, который можно использовать как в школе, так и дома. Перед ребенком ставятся задачи, в которых надо найти давно известные в мире взрослых решения, основанные на таких серьезных понятиях, как алгоритм, рекурсия, дихотомия, стек и т.д. Junior-1 поможет Вашим детям изобрести эти понятия самостоятельно!

FIVE STARS v1.0 - изящный интегрированный пакет, простой и мощный одновременно - специально для Вас и Ваших учеников! Вы научите детей пользоваться электронными таблицами и составите бюджет школы. С Вашей помощью они освоят базы данных и помогут решить Вам все административные проблемы. Вместе с детьми Вы погрузитесь в удивительный мир компьютерной графики и научитесь профессионально работать с клавиатурой. Современный редактор текстов поможет Вам прекрасно оформить любые документы. Five Stars - это маленький мир с большими возможностями!

	Коммерческая цена (USD)	Цена для системы образования (USD)
JUNIOR-1 v2.0	117	57
Часть 1	42	20
Часть 2	48	23
Часть 3	35	17
FIVE STARS v1.0	125	61
Клавиатурный тренажер	19	9
Графический редактор	25	12
Электронные таблицы	32	15
База данных	29	14
Текстовый редактор	32	15
TEACHCAD v3.5	99	49

Все цены указаны в долларах США, но Вы можете заплатить рублями по текущему курсу ММВБ.

Если Вас заинтересовали эти программные средства, обращайтесь, пожалуйста, к нашему дилеру - НЦПСО (Научный центр программных средств обучения).

Адрес: 109004, г.Москва, ул.Б.Коммунистическая, 9А, НЦПСО при МДО.

Телефон: (095) 272 26 71, Сергей Сергеевич Майоров.

р/с 609909 в Гагаринском коммерческом банке г.Москвы МФО 201315.

АСТАВ

Advanced Instructional Software Trading AB

Advanced Instructional Software Trading AB представляет в странах бывшего СССР интересы фирмы IST AB - International Software Trading AB и предлагает программные продукты для системы народного образования.

для Вас!
для Вас!
для Вас!



ИНФОРМАТИКА И ОБРАЗОВАНИЕ

март —
апрель

СОДЕРЖАНИЕ

ОБЩИЕ ВОПРОСЫ

- Краснов А. Система информатизации высшей школы России 3
Ваграменко Я., Королев В. Новые разработки для образо- 5
вания
Конференция-семинар «Телекоммуникации в сфере обра- 7
зования»

КРУГЛЫЙ СТОЛ

- С какого класса преподавать информатику? 8

МЕТОДИКА

- Щеголев А. Информатика и диалектика. (Окончание) 11
Тимашов В., Карасева Т., Гомзякова В. Программа курса
«Основы информатики и вычислительной техники». 20
(Окончание)
Видерхольд А., Митцлафф Х., Грис У. Компьютер в на- 23
чальной школе

НОВЫЕ ПРОЕКТЫ

- Богданова Д., Федосеев А., Христочевский С. Телеком- 27
муникации для образования
Основы экологического земледелия и маркетинга личного 29
и малого фермерского хозяйства

КАБИНЕТ ВТ

- Ковтун М. Программирование в системе Express Pascal. 33
(Окончание)
Стаценко П., Подольский Л., Семионенков М. Интегри-
рованная система программирования Quasic-3 для ДВК-3М,
ДВК-4 и «Электроники-85» 42
Щеголев А. Школьная информатика и язык Пролог 52
Ерохина Е. От логики к программированию (Пролог
в курсе информатики) 57

ДОСКА ОБЪЯВЛЕНИЙ 68

РАЗЛОЖИМ ЗВУК, ВЗВЕСИМ И ИСЧИСЛИМ

- Усенков Д. Музыкальный редактор для Бейсик-Вильнюса 71
БК — 0010.01

Пименов Г., Пименов Д. Четырехголосый музыкальный редактор на Бейсике для БК-0010.01 (БК-11М)	74
Самойлов В. Программирование мелодий на БК-0011М	79
Маслов В. Музыка на Форте	80

ОБМЕН ОПЫТОМ

Рахманкулов Р. Звуковые эффекты на БК-0010	81
Ивашинников С. О создании музыкального оформления	83
Попов Д. Говорящий «Апогей»	83
Лукичев В. Звуковое оформление программ	84
Новиков Ф. Подпрограмма в ПЗУ	86
Николаев В. Рев БК	86

КЛУБ БК

Усенков Д. От БК-0010 до БК-0011М	87
Саяпин А., Вормсбехер В. Знакомьтесь: БК-0011М	93
Конюшенко А. Система машинных команд БК-0010	96
Таланов С. БК-0010 и телетайп	101
Котов Ю. Простое соединение ПЭВМ типа IBM PC и БК-0010	104

КЛУБ «КОРВЕТ»

Кононов С. «ЛИнТех» представляет: Локальная сеть с IBM-совместной головной машиной NET-CP/M версия В-1.0	111
Рогачев А. Связь по магнитофону	116

КЛУБ «АГАТ»

Новиков А. Такой же, только без крыльев. (Окончание)	117
--	-----

ПЕДАГОГИЧЕСКИЙ ОПЫТ

Дементьев В. Сетевая экономическая учебная среда «Капитал»	121
Крылов И. Пакет программ для ПЭВМ «Корвет»	125
Огородников Е., Краснова Р. УПК готовит к профессиям с использованием ЭВМ	126

ИНФОРМАЦИЯ

Справочный листок	128
-------------------	-----

Главный редактор
академик

В. А. МЕЛЬНИКОВ

РЕДАКЦИОННАЯ
КОЛЛЕГИЯ

И. Н. АНТИПОВ
В. Н. АФАНАСЬЕВ
И. М. БОБКО
Б. М. ВАСИЛЬЕВ
(редактор отдела ВТ)

Г. В. ГОДЖЕЛЮ
С. А. ЖДАНОВ
А. Ю. КРАВЦОВА
(зам. главного редактора)

Н. Г. МЕЛЬДИАНОВ
И. С. ОРЕШКОВ
О. К. ПАВЛОВА
А. Ю. УВАРОВ
А. И. ФУРСЕНКО
В. О. ХОРОШИЛОВ

Обложка С. Плюща
Редактор отдела Н. Луцкая
Ведущий редактор
Т. Захарова
Художественный редактор Н. Васильева
Технический редактор Т. Луговская
Корректор И. Богданова

Сдано в набор 31.03.93. Подписано в печать 13.05.93. Формат 70×100 1/16.
Бумага офсетная. Печать офсетная. Усл. печ. л. 10,40. Усл. кр.-отт. 21,80.
Уч.-изд. л. 13,5. Тираж 33 062 экз. Заказ 375. Цена 50 руб.

Почту направлять по адресу:
117485, Москва, а/я 34.
Журнал «Информатика и образование».
Телефон 208-30-78

При перепечатке материалов ссылка на журнал
«Информатика и образование» обязательна.
Ордена Трудового Красного Знамени Чеховский полиграфический
комбинат Министерства печати и информации Российской Федерации
142300, г. Чехов, Московской обл.

А. КРАСНОВ

Российский координационный центр информационных технологий в образовании

Система информатизации высшей школы России

Радикальная теория высшей школы, осуществленная в настоящее время, предполагает опережающее ее развитие по отношению к технической реконструкции производственно-хозяйственного комплекса страны. Обеспечение единства научной и учебной работы и использование в учебном процессе передовых достижений науки и техники позволит высшей школе принять активное участие в научных исследованиях, проектных и конструкторских разработках в интересах государства и общества.

Современный этап развития высшей школы России требует рассматривать ее как целостный социально-экономический объект, результативность и эффективность деятельности которого определяются степенью информатизации учебного процесса и гибкостью системы управления.

Началом отсчета современного этапа информатизации образования можно считать 1988 г., когда под руководством первого заместителя Председателя Гособразования СССР Ф. И. Перегудова впервые в истории бывшего Советского Союза для высших учебных заведений страны за рубежом были закуплены персональные компьютеры фирмы IBM. Появление нового поколения средств вычислительной техники и интенсивное развитие средств телекоммуникации предоставили новые возможности для совершенствования системы образования.

Совершенствование организационного механизма деятельности высшей школы, дальнейший подъем уровня информированности на основе внедрения вычислительной техники в практику управления системой образования, использование новых информационных технологий в процессе обучения — основные внутренние резервы повышения качества подготовки специалистов и эффективности

результатов научных исследований вузов.

В соответствии с этим одним из актуальных направлений деятельности высшей школы является развитие информационной инфраструктуры отрасли — создание региональных (краевых, республиканских, областных, городских, специализированных) центров новых информационных технологий (центры НИТ) и, как следствие, повышение эффективности использования научного и интеллектуального потенциала вузов в развитии информатизации регионов. Эта структура должна стать основой информатизации вузовской науки и образования.

В настоящее время организационная структура отрасли насчитывает 10 региональных, 37 республиканских, краевых, областных, городских и 5 специализированных центров новых информационных технологий, функционирование которых осуществляется в тесном взаимодействии с Российским координационным центром информационных технологий в образовании (Москва). Применительно к настоящим условиям центры НИТ целесообразно рассматривать как гибридные структуры, обеспечивающие поддержку разнообразных функций и услуг для различных потребителей. Существующая иерархическая структура центров НИТ вертикально и горизонтально взаимосвязана информационными и экономическими отношениями. Базовыми организациями центров помимо базовых подразделений вузов (кафедр, лабораторий, вычислительных центров и т. д.) выступают также школы, ПТУ, лицеи, техникумы и т. д. Каждый центр НИТ обладает достаточной организационной и финансовой самостоятельностью для проведения эффективной деятельности в рамках своего региона (края, области, города).

Российский координационный центр ин-

3

формационных технологий в образовании в структуре этой системы является основным связующим звеном, обеспечивающим скоординированную деятельность всей системы в целом. Этот центр отстаивает интересы системы информатизации на уровне Комитета по высшей школе Российской Федерации, а также обеспечивает взаимодействие с другими сторонними организациями, в том числе инофирмами. Являясь головной организацией по Межвузовской научно-технической программе «Развитие информационной среды высшей школы», центр формирует информационную систему Комитета по высшей школе и всей отрасли в целом, которая в ближайшее время позволит получать разностороннюю информацию по деятельности высшей школы, в том числе по электронной почте.

Центры НИТ это:

узловые точки по сбору, обработке, накоплению, анализу и передаче информации о высшей школе, для высшей школы и образования в целом (не исключается оказание этих информационных услуг и для других отраслей);

коллективы преподавателей и ученых, занимающихся разработкой и внедрением информационных технологий не только для системы образования, но и для других отраслей страны;

испытательные и демонстрационные площадки для новой техники и программного продукта, в том числе зарубежных на базе имеющихся в их структуре демонстрационных залов и фондов алгоритмов и программ;

организации, обеспечивающие переподготовку и повышение квалификации кадров в области новых информационных технологий; инструмент по привлечению средств для развития системы образования из других отраслей.

В настоящее время экономической фундаментом деятельности центров НИТ (без учета сторонних источников) базируется на трех основных бюджетных составляющих:

— выделяемый фонд заработной платы профессорско-преподавательского и учебно-вспомогательного персонала;

— выделяемые средства на научно-исследовательские разработки по программам «Информатизация России», «Информатизация образования», «Развитие информационной среды высшей школы», а также средства, получаемые от участия в различных конкурсах;

— первоочередное выделение материально-технического обеспечения.

Разработка и внедрение новых информационных технологий во всех отраслях невозможны без изучения и использования огромного рынка программного продукта, как инструментального, так и прикладного

назначения, в том числе зарубежного производства. Для этих целей в Российском координационном центре информационных технологий в образовании функционирует Отраслевой фонд алгоритмов и программ, имеющий свои филиалы и отделения в центрах НИТ. Этот фонд обеспечивает фондирование (включая госрегистрацию), тиражирование и продажу программных средств, как разработанных вузами и центрами НИТ, так и лицензионно чистых программных продуктов инофирм на дилерских и дистрибуторских правах. В настоящее время в фонде находится около 400 вузовских разработок и свыше 70 лицензионно чистых программных продуктов инофирм, предлагаемых для приобретения различным организациям, в том числе и со скидкой для учебных заведений. В ближайшее время на базе фонда завершится работа по созданию сертификационного центра программных продуктов.

Основным способом общения у центров НИТ в 1992 г. стала электронная почта «RELCOM», адреса которой имеют все центры НИТ и значительное число вузов. Еще большие возможности по информационному обмену (в экономическом плане) представляет созданное для учреждений образования и академической науки ответвление упомянутой сети — «RELARN».

Какую же роль будут играть центры НИТ в развитии и информатизации регионов и страны в целом?

Процесс информатизации в различных регионах России развивается и будет развиваться неравномерно. Новые информационные технологии будут приживаться в первую очередь там, где имеется развитая сеть коммуникаций, налажена сфера информационного обслуживания, действуют службы технического сервиса, где имеются компьютерные и иные клубы, позволяющие пользователям современной техники обмениваться опытом и вырабатывать коллективные решения. И в этом случае центры НИТ должны стать теми точками роста, где сконцентрируется все новое, где будет обеспечено многообразие творческого развития личности. Этот процесс должен управляться и стимулироваться на региональном уровне местными органами управления. Главная задача центров НИТ состоит в том, чтобы выбрать соответствующую модель внедрения НИТ, отвечающую местным условиям, организовать подготовку и переподготовку педагогических кадров, оказать помощь в оснащении учебных заведений необходимыми методическими разработками, техническими и программными средствами, обеспечить их эксплуатацию.

Направленность информационных систем и технологий высшей школы на поддержку формирующихся рыночных структур и отно-

шений является основной чертой современного развития вузов и центров НИТ.

Информатизация образования требует значительных бюджетных средств для подготовки и переподготовки кадров, приобретения, установки и обслуживания средств информатизации, разработки, производства и поставки программных продуктов. Для наиболее эффективного использования научного и интеллектуального потенциала вузов и центров НИТ в развитии информатизации регионов следует использовать сочетание традиционной формы государственного финансирования процесса информатизации с широким привлечением других источников (средств предприятий, местных органов управления и т. д.).

Исходя из этого, первоочередной комплекс задач, возлагаемых на центры НИТ, предполагает решение следующих проблем: проведение научно-исследовательских работ в направлении информатизации обучения, проектирования, организационной деятельности и муниципальной информатики; создание аппаратно-программных и методических комплексов в интересах конкретного региона, разработка методических и нормативных материалов в области новых информационных технологий; организация повышения квалификации и переподготовки специалистов в области новых информационных технологий; производство информационных, консультационных и посреднических услуг; участие в создании территориальных баз данных.

В связи с изменившейся экономической ситуацией, когда высшая школа оказалась в условиях рыночной экономики, в центрах

новых информационных технологий наряду с госбюджетными научно-исследовательскими работами ведутся коммерческие разработки. К практическим результатам работ, направленных на решение региональных проблем, следует отнести работы, выполняемые в нижеперечисленных центрах:

Северный региональный центр НИТ (Петрозаводский государственный университет) — разработка историко-архитектурной базы данных деревянного зодчества Российской севера;

Рязанский областной центр НИТ (Рязанский радиотехнический институт) — разработка средств автоматизации контроля и управления экологическими установками;

Брянский областной центр НИТ (Брянский институт транспортного машиностроения) — создание программного и информационного обеспечения средств НИТ для оценки экологического состояния территории в Российской зоне радиоактивного загрязнения от аварии на Чернобыльской АЭС;

Башкирский республиканский центр НИТ (Уфимский авиационный институт) — разработка информационно-диагностической системы «HOSPITAL» для ожоговых центров и др.

В целях дальнейшего успешного развития создаваемой инфраструктуры информатизации образования и, как следствие, регионов в целом, центры НИТ в 1992 г. учредили ассоциацию «НИТОН». Созданная ассоциация может в перспективе оказать значительную экономическую поддержку развивающимся центрам НИТ.

Я. ВАГРАМЕНКО, В. КОРОЛЕВ

Российский центр информатизации образования

Новые разработки для образования

В 1992 г. по заказу Министерства образования РФ был выполнен ряд исследований в рамках программы информатизации образования. Координатором пяти проектов этой программы стал Российский центр информатизации образования (РОСЦИО). Ниже коротко представлены результаты этих работ.

Разработке и созданию банка данных о техническом, программном и методическом обеспечении информатизации в образовании посвящено исследование в рамках проекта «Создание информационного банка о потенциальных возможностях развития сферы образования России в области информатизации» (РОСЦИО, Москва, руководители Ваг-

раменко Я. А., Колыхалов П. И.). В результате работы создана концепция информационного банка о программно-аппаратных, методических средствах, научных исследованиях, программы подготовки кадров, связанных с применением и использованием новых информационных технологий в образовании. Авторы создали программный комплекс и наполнили информацией базы данных, реализовав предложенную концепцию. На сегодняшний день банк содержит сведения по педагогическим вузам республики, школам, имеющим «пилотные классы», и институтам повышения квалификации работников образования (ИПКРО). Банк предоставляет возможность:

получить информацию для анализа уровня информатизации образовательных учреждений России;

получить информацию для анализа парка компьютеров, программных и методических средств, используемых в сфере образования;

ускорить информационный обмен данными об аппаратном, программном и методическом обеспечении в системе образования.

Один из проектов был нацелен на построение совместной учебной работы педагогов и учащихся с использованием компьютерной коммуникации. Работа выполнена в Московском институте развития образовательных систем (МИРОС, Москва, руководитель Уваров А. Ю.). Результатом исследований стало:

построение каталога совместных учебных проектов, выполняемых с использованием средств компьютерной коммуникации;

построение каталога компьютерных сетей, предоставляющих услуги для их использования в учебном процессе в общеобразовательной школе;

описание опыта реального использования учебной компьютерной сети в практике работы одной из московских школ (на примере одной из известных учебных сетей: Kidnet, Global, Classroom, ATT Learning Circle); построение системы моделей учебной работы учащихся общеобразовательных школ совместно с партнерами в компьютерной сети по методу совместных учебных проектов.

Авторы подготовили пособие для педагогов «Компьютерная коммуникация в школе».

Целью другой работы (РОСЦИО, Москва, руководитель Григорьев С. Г.) являлась концепция выбора средств вычислительной техники учебного назначения для учреждений образования в России. Авторами предложены критерии отбора учебной вычислительной техники, методика системного анализа вариантов учебной вычислительной техники, рекомендации по отбору варианта средств вычислительной техники учебного назначения и программа обеспечения высших и средних образовательных учебных заведений России средствами вычислительной техники. Следует также отметить полный и критический анализ типов средств вычислительной техники и подробное рассмотрение возможных вариантов программно-методического обеспечения вычислительной техники, предназначенной для высших и средних образовательных учебных заведений России.

Для информатизации сферы управления образованием несомненный интерес представляет комплекс работ «Информатизация системы управления в сфере образования» (Информационно-консультационный аналитический центр «ИКА», Москва, руководитель

Жеребцов А. В.). Авторами получены следующие результаты:

концепция информатизации управления в сфере образования и программно-методическое обеспечение единого информационного пространства управления отраслью;

пакет программ и данных для информационного обеспечения системы управления. В итоге работы создано рабочее место министра образования России с учетом его требований. Выполнено проектирование, разработка и опытная эксплуатация мини-локальной сети «Министр — секретариат — управление» с возможностью работы в информационной среде с построением информационных баз по входящей и исходящей документации, нормативной документации и автоматизации ввода информации с первичных документов. Проведена разработка и опытная эксплуатация компьютерной связи «Министерство — регион» на базе Воронежской области с установкой в Воронежской области необходимых программных средств, в том числе и обеспечивающих деятельность органа управления образованием. Определены первоочередные и перспективные задачи, стоящие перед министерством и территориальными органами управления по информатизации их деятельности.

Еще одна работа выполнена в Уральском компьютерном центре (УрКЦ, Екатеринбург, руководители Нигаев А. Н., Маршалкин А. П.). Эти исследования продолжают серию работ, посвященных применению банков данных в образовании. Проект «Разработка банка данных по дефектологии» касается создания комплекса диагностических программ на базе ПК для своевременного выявления отклонений в психофизическом и интеллектуальном развитии детей с целью оптимизации коррекционной работы и развития творческих способностей ребенка. В результате авторам удалось внедрить в широкую практическую деятельность специальных детских учреждений адаптированные для компьютерного использования известные и оригинальные психофизиологические методы исследования, позволяющие максимально объективизировать диагностическо-коррекционную работу. Также удалось сократить время исследований и провести системный анализ динамики развития ребенка.

Цикл работ, представленных выше, заметно восполняет недостаток средств прикладной информатики для образования, который придется устранить в ближайшие годы. Для продолжения работ получен существенный задел.

Конференция-семинар «Телекоммуникации в сфере образования»

Необходимое условие успешного осуществления реформы образования в России — создание единого информационного пространства образовательных учреждений. Только обеспечив доступность информации всем образовательным учреждениям, можно обеспечить широкое распространение педагогических инноваций, расширить учебно-методический арсенал учителя, дать ему возможность творчески строить учебный процесс.

Наиболее естественное средство решения указанной задачи — создание образовательной телекоммуникационной сети. Накопленный за последние годы опыт использования средств телекоммуникации в образовании (в частности, многие вопросы нашли отражение в публикациях на страницах журнала «Информатика и образование») сделал это реальным в масштабах страны.

Конференция в г. Рязани стала первым шагом, предпринятым министерством образования РФ для объединения усилий разработчиков телекоммуникационных сетей в сфере образования и создания Российской образовательной телекоммуникационной сети.

Создаваемая сеть должна быть некоммерческой. Все затраты ее абонентов не должны превышать расходы на оплату каналов связи и содержания Host-компьютеров.

На конференции была принята следующая стратегия построения сети:

На первом этапе образовательные учреждения, имеющие соответствующие технические и кадровые возможности, организуют узлы сети, к которым другие образовательные учреждения подключаются в качестве абонентов. Обмен информацией между узлами сети осуществляется на основе каналов связи существующих сетей (в настоящее время для этих целей решено использовать образовательное подмножество сети Релком — Реларн). По мере накопления опыта работы в сети будут выделены узлы, которые приступят к обслуживанию включающихся абонентов.

В ближайшее время предполагается осуществить подключение существующих в отрасли баз данных к создаваемой сети и сделать их информацию доступной всем абонентам. По мере развития сети количество баз данных и их информационное наполнение будет увеличиваться.

Координацию работ по развитию сети, ее программному обеспечению, аппаратному оснащению и т. п. поручено вести Центру,

создаваемому на базе Рязанского педагогического института.

Сеть предполагается использовать для решения следующих задач:

— обеспечение обмена между образовательными учреждениями и органами управления образованием всеми видами служебной и управленческой информации;

— предоставление всем образовательным учреждениям возможности получения информации из создаваемых и подключаемых к сети баз данных (нормативных документов, учебно-методических материалов, учебных компьютерных программ и пр.);

— совместного осуществления образовательными учреждениями всех типов учебных проектов;

— проведение конференций по актуальным вопросам развития образования;

— создание информационного агентства в сфере образования.

По мере развития сети список решаемых ею задач будет расширен. Ниже приводятся основные положения решения конференции.

Участники совещания пришли к единодушному выводу о том, что в настоящее время созрела необходимость создания единой Российской образовательной телекоммуникационной сети (РОС).

Создание единой Российской образовательной телекоммуникационной сети (РОС):

— является важным фактором сохранения единого образовательного пространства в России;

— обеспечивает основу необходимого развития содержания, методов и организационных форм обучения;

— позволяет перестроить систему управления образованием с учетом процессов автономизации и регионализации образовательных систем.

Участники совещания констатируют, что современные средства вычислительной техники и уровень подготовленности педагогических кадров позволяют приступить к практическому решению данной проблемы.

По мнению участников совещания, РОС должна представлять собой систему, создающуюся на базе образовательных учреждений России, действующую в интересах образования и открытую для всех организаций и лиц, содействующих развитию образования.

РОС должна представлять собой распределенную структуру с равноправными узлами, строиться с учетом лучших мировых образ-

цов в области создания образовательных телекоммуникационных сетей, позволять связываться с образовательными учреждениями за рубежом, обеспечивать минимальный уровень затрат для ее поддержания и использования. РОС должна создаваться как эволюционирующая система, вырастающая по мере появления у образовательных учреждений потребности в соответствующих коммуникационных услугах. При создании РОС рекомендуется опираться на имеющийся потенциал педагогических вузов и научных подразделений РФ и других учебных и научных организаций, работающих в сфере образования.

Участники совещания считают целесообразным:

1. Просить Министерство образования РФ предусмотреть развертывание работ по созданию РОС в качестве приоритетной зада-

чи Государственной научно-технической программы «Развитие образования в России» («Информатизация образования»); выйти с ходатайством в соответствующие инстанции о предоставлении льготных тарифов на услуги связи для РОС; содействовать включению РОС в международные образовательные компьютерные сети.

2. Создать на базе Рязанского государственного педагогического института им. С. А. Есенина «Центр компьютерных коммуникаций в образовании», поручив ему координацию работ по созданию РОС.

3. Рекомендовать органам управления образования, педагогическим вузам, научным подразделениям Министерства образования РФ принять участие в создании РОС.

4. Предложить организациям и лицам, заинтересованным в развитии образования, включиться в работу по созданию РОС.

8 КРУГЛЫЙ СТОЛ

С какого класса преподавать информатику?

Осенью 1992 г. в Республиканском институте повышения квалификации работников образования состоялся круглый стол, в котором приняли участие заведующие кабинетами информатики институтов повышения квалификации из разных городов России, представители редакции журнала «Информатика и образование», автор одного из учебников по информатике А. Г. Гейн. Обсуждались две темы: «С какого класса преподавать курс ОИВТ?» и «Информатика в начальной школе». Ниже публикуются фрагменты из выступлений участников.

Грушанская Е. А. (*г. Иваново.*) Я считаю, что темы, предложенные для обсуждения, достаточно актуальны. Идет время, но к началу очередного учебного года мы так и не узнаем ни планов министерства образования РФ, ни ясных и четко сформулированных положений о преподавании информатики, в частности, с 8 класса, в начальной школе, не видим новых учебников и методических пособий. Нужно учесть, что во многих школах уже практически перешли к ведению курса «Основы информатики и вычислительной техники» с 8 класса. Например, наш регион давно к этому готов, но все тормозится отсутствием учебных планов, учебников, соответствующего программного обеспечения. Кроме того, разнотипность техники создает значительные трудности. Сегодня нам предлагают программно-методические комплексы по инфор-

матике, но инициатива исходит от авторских коллективов, а не от министерства. Почему эти комплексы не распространяются централизованно? Ведь при таком подходе одна школа сможет их приобрести, а другая — нет. В результате заведующий кабинетом в институте усовершенствования занимается и закупкой программ, и координацией работы. Все это требует огромных усилий, времени, энтузиазма. Программы стоят дорого, и институт усовершенствования зачастую не в состоянии их приобрести. Это приводит к разделению школ на бедные и богатые.

Есть еще одна проблема: отсутствие налаженной системы обмена опытом, хотя процессы информатизации в регионах развиваются.

Герасимова Т. Н. (*г. Тверь.*) В нашей области до 15 типов техники, много программного обеспечения и преподавание ведется по трем учебникам. Поэтому мы не берем на себя координацию по сбору и внедрению программ — это нецелесообразно, да и неподъемно. Перед нами стоит финансовая проблема: где взять деньги на приобретение программного обеспечения? Многие учителя области обращаются в методический кабинет института с предложением начать преподавать курс информатики с 8 класса. Мы составили программу, утвердили ее, и те, кто сказал заинтересованность, по ней работают. Имеются апробированные и хорошо себя зарекомендовавшие методики и про-

граммы. Информатика в начальной школе вводится пока в форме кружковой работы.

Елисеева Е. С. (г. Саратов). Преподавание информатики в г. Саратове имеет давнюю традицию — ведется уже около 30 лет. С 7 класса в физико-математической школе при Саратовском государственном университете на ЭВМ «Урал-2» школьникам читались курсы, связанные с вычислительной техникой и программированием. В лицеях, гимназиях, других физико-математических школах области ввели такие курсы с 8 класса, а в 1992/93 учебном году многие обычные школы перешли на преподавание информатики с 8 класса. Но в этом случае возникает проблема: что делать с учениками в 10 и 11 классах, если начать преподавание по имеющимся учебникам? На мой взгляд, авторы еще не готовы удовлетворительно ответить на этот вопрос. Кроме того, я совершенно уверена, что учебник и программное обеспечение разделять нельзя. Это единый комплекс. Область готова перейти к преподаванию информатики с 8 класса, но при наличии в достаточном количестве необходимого методического сопровождения, учебников, программных средств и т. д.

«ИНФО». Сколько сейчас стоит программная поддержка учебников?

Меланина Г. М.

Это очень большой вопрос. Средняя цена пакета 2,5—3 тыс. руб., а некоторые стоят и 5 тыс. руб. (Цены на осень 1992 г.—Прим. ред.). Для школы это все-таки большие деньги. Некоторые сельские школы вынуждены буквально торговать картошкой для того, чтобы собрать такую сумму. Бюджетных средств не хватает. Сегодня назрел переход к преподаванию информатики с 8 класса. В 10 и 11 классах изучать этот предмет уже поздно. Но нужны новые, соответствующие учебники и программные средства, а их приобретение сегодня практически поставлено на псевдорыночную, коммерческую основу. Все-таки этот вопрос с приобретением необходимых учебных материалов должен решаться централизованно.

Гейн А. Г. (г. Екатеринбург). Продаются программно-методические комплексы не авторы, а те учреждения, в которых они работают. Мы действительно получаем процент от продаж, если заключен соответствующий договор. Проблема состоит в том, что создание такого комплекса стоит очень дорого для разработчиков и передавать в школы все это бесплатно никто не согласится. Однако наш коллектив передал 10 000 копий программ для учебника в министерство образования РФ. Всем желающим мы предлагаем обращаться непосредственно к нам. Естественно, централизо-

ванное тиражирование было бы предпочтительнее.

Чванова Н. С. (г. Тамбов). Проблема преподавания информатики с 8 класса должна быть решена на официальном уровне и как можно быстрее. В этом нет никаких сомнений. Мы и так уже опоздали, потому что этот процесс стихийно идет. Подготовка кадров, разработка методик — это решаемые вопросы.

Елисеева Е. С. (г. Саратов). Почти все здесь присутствующие знают, и мы это с интересом обсуждаем, что началось преподавание информатики и в начальной школе. Если мы опоздаем так же, как и с восьмым классом, то не исключено, что процесс пойдет так же стихийно, и в изуродованной форме.

Тимашов В. П. (РИПКРО). Дело в том, что в настоящее время администрация школы имеет право самостоятельно утверждать учебные планы и вводить новые дисциплины, в том числе и информатику в начальной школе. Лично я считаю, что пока нет тщательно разработанной концепции, рекомендаций врачей, эргономистов и психологов ни в коем случае нельзя, даже в экспериментальном порядке, усаживать детишек за компьютер. Иначе через 2—3 года мы столкнемся с проблемами здоровья наших детей.

Солдатова Н. М. (г. Владивосток). Весной 1992 г. мы провели в нашем крае семинары на эти же темы. Учителя на местах создают программы, разрабатывают методики, экспериментируют. То же самое происходит и в Хабаровском крае, и в Южно-Сахалинской области. Так что запретить что-либо просто нельзя. На мой взгляд, министерство образования РФ должно уже сейчас осознать всю значимость вставшей проблемы и начинать работу в этом направлении. Информатика в младших классах стучится в дверь. Нельзя не учитывать и зарубежный опыт.

Хотелось бы отметить еще чрезвычайно важную проблему, имеющую отношение к этому вопросу, — текучесть кадров. Занимаются информатикой в младших классах энтузиасты. Поэтому при смене учителя — беда. Пока не будет единой программы, не будет у ребят систематизированных знаний.

Егоров А. П. (г. Йошкар-Ола). Запретить преподавание информатики в младших классах можно на основании «Закона об образовании». Если соответствующая комиссия считает, что данная техника вредна для здоровья ребенка, то директор школы и учителя по существующему законодательству могут быть привлечены к ответственности, даже к судебной. Да и информатику с 8 класса можно вводить при на-

личии нормативных документов. Для введения или отмены какого-либо предмета должны быть свои основания.

Волынский О. В. (г. Рязань). И в нашей области уже преподают информатику с 8 класса. В управлении народного образования этот вопрос решен, утверждены планы. Но санэпидемстанция закрывает дисплейные классы, ссылаясь на письмо от 20.02.92 г., в котором указан перечень мониторов, запрещенных к использованию в учебно-воспитательном процессе. Кроме того, существующие нормы по площадям, по кубатуре и т. п. уже давно не пересматривались. Не публикуются результаты эргономических исследований, рекомендации гигиенистов. А ведь этому вопросу при введении курса информатики уделялось значительное внимание.

Елисеева Е. С. (г. Саратов). Назрела и ждет своего часа проблема использования средств информатики в других предметах. Здесь тоже нужны решения со стороны органов управления. Учителя-предметники хотят использовать вычислительную технику, размещенную, как правило, в кабинете вычислительной техники, для проведения своих уроков. Кто, как не учитель информатики, может помочь им хотя бы вначале и с консультациями по использованию ВТ, и с подбором необходимых программных средств, и с организацией урока?

Грушанская Е. А. (г. Иваново). Учителю-предметнику оплачивают часы, а кто бу-

дет оплачивать эту работу учителю информатики? Некоторые директора школ оплачивают такие часы как факультативную или кружковую работу, но это не выход из положения. Подводя итоги нашего круглого стола, мне хотелось бы сформулировать общие для нас, как мне кажется, вопросы и попросить представителей министерства образования РФ на них ответить. Есть ли сегодня в России концепция информатизации образования? С какого класса преподавать информатику? Как будет решаться проблема с распространением учебников и программно-методического обеспечения? На какую технику ориентироваться в дальнейшей работе? На какие средства школа должна приобретать технику и программы? Как должен оплачиваться труд учителя информатики с учетом его специфики? Только когда будут даны ясные ответы на эти и другие вопросы можно, на мой взгляд, говорить о введении информатики с первого класса, о компьютеризации школы, о современном использовании информационных технологий в образовании.

От редакции. Мы попросили главного специалиста ведущего отдела информатизации образования министерства образования РФ Самовольнову Л. Е. прокомментировать материалы круглого стола. Редакция предполагает опубликовать комментарии в ближайшем номере.

Наша специализация - разработка и распространение аппаратных и программных средств для БК0010(-01), БК0011М

Фирма Альтек предлагает:

- * средства для подключения дисководов к БК0010, БК0011М;
- операционную систему ANDOS для работы с дискетами в формате IBM на БК;
- издательские системы и другие программные средства для работы с различными принтерами;
- * авторское программное обеспечение (системные, прикладные, обучающие и развивающие программы);

А также:

- компьютеры БК0010, БК0011М
- мониторы
- дисководы
- принтеры
- контроллеры дисководов
- дополнительная память 8 и 16 КБайт

Наш адрес: 109444, Москва, а/я 38. Тел.: (095) 377-7436.

А. ЩЕГОЛЕВ

Информатика и диалектика

8-й класс

Достигнутое в 7-м классе представление о базах знаний как об удачной модели человеческого мышления разрушается первыми же (специально подобранными) примерами баз знаний в 8-м классе. Сравнение мышления человека и имитирующих его баз знаний оказывается не в пользу последних. Яркий характер «кризиса» заставляет задуматься над методологическим вопросом: каковы границы применимости («время жизни») каждой модели: ведь ни одна модель не отражает моделируемый объект во всей полноте (во всем многообразии аспектов).

Результатом поиска ответа на вопрос о выходе из тупиковой ситуации является формирование представлений о взаимном соответствии структуры процесса и структуры результата этого процесса, иначе говоря, о диалектике исторического и логического. В частности, структурная неоднородность результата деятельности позволяет делать выводы о «неравномерном» (не только непрерывном, но и дискретном) характере самого процесса.

Отсюда возникает важный вывод о том, что следующей в последовательности моделей мышления человека должна стать «трех-этажная» структура, в которой над непрерывно пополняемой базой знаний надстраивается «этаж», отвечающий за своевременные дискретные (качественные) изменения баз знаний. В известном смысле это эффект увеличения размерности системы: формально-логическая («плоская») модель мышления (база знаний) обобщается

до содержательной («пространственной»), являющей собой последовательность «плоских» моделей. 11

Особое внимание следует обратить на проявление схожих принципов в различных предметных областях: искусстве, биологии и т. д.

Тематическое планирование (2 ч/нед×34 нед=68 ч)

1. «Кризис» представлений о базах знаний (5 ч)

«Воспоминания» о базах знаний как модели человеческого мышления. Факты, правила, вопросы; процесс рассуждения. Особенности процедуры поиска ответа на вопросы, требующие ответа «да» — «нет» (поиск проводится лишь до первого решения — положительного ответа на вопрос).

Примеры «аномальных» баз знаний:

- * с тавтологией;
- * со скрытой (замаскированной) тавтологией;

- * со взаимно обратными правилами.

«Защипывание» процесса поиска ответа на вопросы к «аномальным» базам знаний.

Порядок расположения фактов и правил в подобных базах знаний существенно влияет на получаемые ответы на вопросы (в особенности это касается вопросов, требующих ответа «да» — «нет»)!

Однако для мышления человека, очевидно, ни тавтологии, ни взаимно обратные правила не представляют затруднений.

Вывод: мы обнаружили, в чем базы знаний как модель человеческого мышления проигрывают мышлению человека.

2. По системе Станиславского или... (8 ч)

На одном-двух уроках проводятся «артистические этюды»: практикум с последова-

* Окончание. Начало см.: Информатика и образование. — 1993. — № 1.

тельным вовлечением всего класса в процесс разыгрывания сцен:

а) из школьной жизни (часто встречающиеся в практике школьников ситуации, не требующие существенной смены обстоятельств и серьезного ролевого перевоплощения);

б) с существенным изменением (по сравнению с привычными) времени, характера ситуаций и других обстоятельств по сравнению с привычными житейскими «ролями» школьников.

С организационно-методической точки зрения «показы» проводятся следующим образом.

Преподаватель заранее предупреждает, что в работе примет участие каждый ученик. Для экономии времени и большей психологической комфортности школьников в каждом «показе» могут принимать участие 4—5 человек.

Преподаватель дает «вводную», подробно разъясняя не только внешний антураж предлагаемой ситуации, но и внутренние мотивы, движущие персонажем. При этом детальное описание ситуации и внутреннего мира персонажа не включает описания того, какие именно конкретные поступки должны быть персонажем проделаны.

При этом нужно быть готовым к тому, что результаты отдельных «показов» (главным образом п. «б») будут очевидно неудачными с артистической точки зрения. Поэтому персональное оценивание допускается только в случае явных артистических удач; не оправдавшиеся «воплощения» требуют обобщенного заключения (без «персоналий»).

На следующем занятии необходимо провести анализ просмотров. Основные выводы, на которых рекомендуется остановиться:

а) изображать кого-либо не так-то легко (несмотря на кажущуюся банальность, этот вывод достаточно важен);

б) степень перевоплощения у разных людей различна и зависит от меры артистического таланта;

в) изображать человека, похожего на себя в привычных обстоятельствах, несравненно проще, чем перевоплощаться в образ, в котором многое недостаточно знакомо (включая обстоятельства, в которых «персонажу» приходится действовать).

Проблема: по-настоящему артистически одаренных людей не так много. Как поставить спектакль (или вообще организовать совместную творческую деятельность людей), располагая участниками разной степени одаренности?

Возможны два основных пути решения этой проблемы:

1) («жесткий») — четко предписать каждому человеку, какие действия и в какой последовательности надлежит проделать. Применительно к театральной постановке это означает, что должен быть расписан буквально каждый шаг, каждая смена интонации артиста.

Однако (если даже это удалось бы в полной мере осуществить) всякая «излишне жесткая конструкция» недостаточно жизнеспособна, поскольку не может адаптироваться (приспособиться) даже к небольшим изменениям внешних условий;

2) (предложенный К. С. Станиславским и получивший название системы) не предписывать выполнять поступки (действия) в их жесткой последовательности, а объяснять логику деятельности персонажа. При этом нужные «движения» артист находит сам. Разумеется, при таком подходе сообщить придется гораздо больше. Очевидно, второй подход предполагает более высокий уровень общения с человеком.

Пример для иллюстрации этих двух подходов в житейской практике: объяснить человеку, как добраться до нужного объекта в городе. Здесь опять же возможны два способа.

Первый заключается в том, чтобы дать алгоритм пути. Вводится понятие алгоритма.

Алгоритм — это описание того, какие действия и в какой последовательности нужно проделать для решения поставленной задачи (достижения определенной цели).

Школьникам из собственного опыта известно, что человек, которому объяснили (в виде алгоритма), как добраться до нужного места, как правило, редко находит это место без дополнительных расспросов и уточнений.

Однако возможен и другой способ объяснения. Он заключается в том, чтобы нарисовать схему даже не пути следования, а взаимного расположения объектов (домов, улиц, перекрестков, и т. п.). Этот способ гораздо практичнее именно благодаря более высокому (т. е. более близкому для человека и потому более естественному для него) уровню, чем алгоритмический.

Точно так же и первый способ работы режиссера с артистами может быть назван алгоритмическим, а система Станиславского — способом более высокого (более близкого к творческим качествам человека) уровня.

Возвращаясь к базам знаний, отметим, что записи на Прологе не являются алгоритмами. Ведь база знаний описывает не действия в их последовательности, а структуру объекта. В этом смысле база знаний подобна второму способу представления (т. е. более высокого, чем алгоритмический,

уровня). Однако, когда возникает вопрос к базе знаний, для ответа на него выстраивается последовательность действий в зависимости от конкретности этого вопроса и от структуры объекта, описанной в базе знаний.

Именно нестыковка, несоответствие свойств объектов, отраженных в базах знаний, и порождаемых «зацикливающихся» алгоритмов привлекли наше внимание как «кризис» представлений о базах знаний. Поэтому необходимо разобраться, как связаны свойства баз знаний — вопросов к ним с теми действиями, которые будут проделаны при ответе на вопрос.

3. Деревья — описания действий (10 ч)

Понятие дерева как:

1) графического представления всевозможных состояний;

2) последовательности «обхода» этих состояний.

Напоминание задач из 6-го класса: «Угадай номер квартиры» и «Найди фальшивую монету». Деревья поиска решения этих задач. Сопоставление вида деревьев (двоичного и троичного). Обсуждение.

Дерево игры. Дерево игры в «крестики-нолики». Выигрыш как искомое решение. Попытка создания дерева игры в шахматы для результативной игры. Комбинаторное «проклятие». Вывод: человек (да и компьютерные программы) играют в шахматы не так.

Дерево поиска решения, порождаемое базой знаний и вопросом к ней. Порядок «обхода» дерева — перебор с возвратом. Примеры. Итак, дерево — это наглядный алгоритм поиска решения. Связь порядка расположения фактов и правил в базе знаний с последовательностью действий при поиске ответа на вопрос.

4. Списки на Прологе (10 ч)

Понятие списка как упорядоченного множества констант. Обозначение списков.

Встроенная операция над списками — «отсечение головы». Голова и хвост списка.

Рекурсия как средство обработки списков. Построение функций определения принадлежности элемента к списку (на примере развития базы знаний «Успеваемость» и аналогичных), общего количества элементов и количества данных элементов в списке и т. п.

Деревья вывода рекурсивных построений. Последовательность «обхода» дерева. Решение и «тупики».

5. Деревья и «аномальные» базы знаний (10 ч)

Как выглядит «зацикливание» на дереве вывода решений? Бесконечно растущие «вниз» «ветви» (поддеревья). Компьютер проводит рассуждения путем перебора с воз-

вратом всех фактов и правил, которые имеются в наличии в базе знаний.

Отметим, что бесконечное «зацикливание» при этом не намного хуже перебора конечного, но очень большого числа вариантов, большая часть которых заведомо не содержит решений. Итак, с точки зрения деревьев (процедуры логического вывода) «неприятности» порождены необходимостью огромного (может быть, бесконечного) перебора, возникающего в одном или нескольких поддеревьях.

Конструктивная идея: нельзя ли «отсечь» заведомо бесперспективные в смысле поиска решений, но громоздкие (конечные и бесконечные) поддеревья и оставить для перебора обозримое пространство исходов (т. е. опять же дерево)?

Примеры последовательного уточнения задач и внесения соответствующих качественных изменений в базу знаний. При этом налицо замена предикатов и увеличение «размерности» системы.

Таким образом, главный вывод заключается в следующем: каждая база знаний (как отражение, мысленная модель мира) со временем неизбежно устаревает. При этом никакими умозрительными логическими выкладками не удастся привести эту модель в соответствие с объективной реальностью (материальной и духовной). Необходимо периодически проводить эксперимент, т. е. спроектированное воздействие на объективную реальность и получение реакции (отклика) на это воздействие. Именно это и приводит к периодическим «переосмыслениям» базы знаний.

Поэтому, если мы хотим уточнить модель человеческого мышления, нам необходимо описать механизм, обуславливающий последовательность модификаций баз знаний, связанных с проведением экспериментов. Условно говоря, нам необходимо над имеющимися «двумя этажами» базы знаний надстроить «третий этаж» (метод познания), который определяет методологию познания в целом, т. е. процесса периодического качественного уточнения базы знаний в результате проведения экспериментов. В отличие от непрерывного пополнения базы знаний фактами и правилами этот процесс — дискретный, т. е. скачкообразный.

База знаний представляет собой формально-логическую модель человеческого мышления. Всякая формально-логическая модель ограничена. Однако чередование, смена формально-логических моделей всякий раз после уточняющего «столкновения» с объективной реальностью порождает последовательность таких моделей, которая и является как бы содержательно-логической моделью мира.

6. Деятельность и результат (5 ч)

Соответствие структуры деятельности и структуры результата этой деятельности.

Для чего нужно изучать историю? Естественно, не для зубрежки дат. Знание предистории может дать ответ на вопрос о современном состоянии, поскольку оно предопределено историей развития. Кроме того, можно в определенной мере прогнозировать будущее, поскольку в некоторых аспектах развитие событий «повторяется».

Взаимосвязь онтогенеза и филогенеза в биологии (биогенетический закон). Для чего необходимо конспективное (I) прохождение каждым отдельным индивидом всей истории развития вида? Не проще ли сразу сформулировать представителя вида в его современном состоянии? По-видимому, не проще, и «проитривание» предистории — наиболее экономный путь формирования.

Почему предистория вида разворачивается «конспективно»? В процессе онтогенеза происходит повторение только тех признаков, которые вносят свой вклад в формирование необходимых подсистем нового организма. При этом заметны как плавные (эволюционные) изменения, так и скачкообразные превращения их в новое качество (по признакам напоминающее один из исторически предшествующих видов).

Отметим, что жизнь в окружающем нас мире, несмотря на вероятностный (как нам уже известно, достаточно мощный) механизм естественного отбора, состоит именно из дискретных видов, а не из «плавной», непрерывной «палитры» существ с различным соотношением признаков. Сопоставление картины мира «по вертикали» (историческое) и «по горизонтали» (логическое).

7. Алгоритмический практикум (10 ч)

Деятельность и результат. Взаимное соответствие: однозначно ли влияет структура процесса на структуру результата и наоборот?

Алгоритмический практикум в среде Пролога. Управление процессом поиска результата. Операция «Отсечение» (cut). Примеры последовательного уточнения задачи и применения «отсечения».

Чередование моделей. Знание правильное и неправильное. «Динамический» характер моделирования. Принцип математической индукции с точки зрения взаимосвязи процесса и его результата. Принцип записи информации в ДНК. Сравнительная компактность представлений алгоритма и описания структуры объекта.

Резерв времени (за год) — 10 ч

Инварианты успешного завершения курса 8-го класса

К концу обучения в 8-м классе учащиеся должны обладать следующими представлениями, закрепленными в практической деятельности (по разделам).

1. Модели мышления человека («три этажа»).

Явное расхождение в результатах решения одной и той же задачи человеком и Пролог-машиной свидетельствует о несостоятельности «двухэтажной» (факты и правила) модели мышления человека.

Понятие алгоритма. «Дерево игры»; «дерево» как способы представления алгоритмов.

Дерево поиска ответа на вопрос к базе знаний может содержать бесконечные (или конечные, но бесперспективно громоздкие) ветви. Однако удаление («отсечение») заведомо бесперспективных в смысле наличия решения ветвей невозможно без возврата (и изменения) самой постановки задачи.

Таким образом, нельзя создать базу знаний (формально-логическую модель, картину мира) вообще; при ее создании необходимо учитывать, какого рода задачи будут решаться при обращении к ней. А поскольку классы задач, решаемых человеком, существенно изменяются в продолжение его жизни, это значит, что «базы знаний» (модели мира) человека также должны периодически претерпевать существенные (качественные) изменения.

Таким образом, в качестве более совершенной модели мышления человека разумно принять последовательность баз знаний. Развиваясь эволюционно (пополняясь фактами и правилами), каждая база знаний периодически претерпевает качественные изменения. Возвращаясь к «двухэтажному» представлению, мы можем усовершенствовать эту модель «надстраиванием» третьего этажа, «ответственного» за периодическое качественное обновление баз знаний. Этот этаж мы условно называем «метод познания».

Таким образом, несмотря на то что всякая модель мира неполна (не вполне соответствует действительности и должна периодически уточняться), тем не менее имеет смысл говорить о знании правильном и неправильном (соответствующем или не соответствующем направленности последовательности таких моделей).

2. Динамический Пролог-практикум.

Списки на Прологе. «Отсечение головы и хвоста» списка. Процедура рекурсивной обработки списков.

Дерево поиска решения, порождаемое базой знаний и вопросом к ней. Влияние

порядка расположения фактов и правил в базе знаний на процесс поиска решения и результат этого процесса.

Операция «отсечение» («cut») — управление процессом поиска решения. Зависимость базы знаний от характера решаемой задачи (множества вопросов, которые могут быть заданы к этой базе знаний).

3. Творчество с точки зрения взаимосвязи исторического и логического.

Проблема организации коллективной творческой деятельности людей. Противоречие между единой целью (желаемым результатом) и творческим (вариативным) характером деятельности отдельных участников.

Два основных способа согласования деятельности: жесткий (алгоритмический) и описательный (декларативный). Декларативный способ более емкий (информативен), алгоритмический — менее избыточен.

Базы знаний — пример декларативного описания. База знаний описывает структуру объекта (статика); при возникновении же вопроса к базе знаний возникает динамический процесс (процедура ответа на вопрос). Алгоритмы подобных процессов неявно рассматривались еще в 7-м классе.

Наиболее существенный вопрос: каким образом соотносятся структура процесса (формально описываемая алгоритмом) и структура результата этого процесса (описываемая декларативно, например, базой знаний)?

На примере процесса изменения модели

мира человека можно выявить существенные черты этого соответствия. Процесс развития смешанный: участки количественного накопления сменяют скачки качественных изменений. Эти последовательно обретаемые качества порождают наиболее существенные «сгустки» в структуре результата данного процесса.

Творческая деятельность характеризуется достаточно закономерным результатом, но (в отличие от деятельности рутинной, нетворческой) самый процесс творческой работы не предопределен заранее и не является закономерным.

Рекомендуемая литература

По 8-му классу

1. Григорьев С. Г. Графические средства системы ПРОЛОГ-Д // Информатика и образование. — 1990. — № 6.
2. Григорьев С. Г. Обработка списков на Прологе-Д // Информатика и образование. — 1991. — № 1.
3. Братко И. Программирование на языке ПРОЛОГ для искусственного интеллекта. — М.: Мир, 1990.
4. Мичи Д., Джонсон Р. Компьютер-творец. — М.: Мир, 1987.
5. Станиславский К. С. Моя жизнь в искусстве. — М.: Искусство, 1983.
6. Станиславский К. С. Работа актера над собой. — М.: Искусство, 1985.
7. Станиславский К. С. Из записных книжек (в 2-х т.). — М.: ВТО, 1986.
8. Покровский Б. А. Ступени профессии. — М.: ВТО, 1984.

15



9-й класс

Этот год является заключительным не только для изучения курса информатики, но также и для всего базового образования школьника. Настало время приступить к системному осмыслению методологии научного познания, которая неявно применялась в курсе информатики на протяжении трех предыдущих лет изучения курса.

Целесообразно проведение занятий нетрадиционным образом: системой семинаров, «круглых столов» и т. п. Безусловно необходимым является соблюдение следующих условий:

применение разного рода наглядных схем для визуализации постановок и решения задач. Не следует опасаться упреков в «огрублении» сущности исследуемых явлений;

важно только постоянно обращать внимание учеников на то, каковы границы применимости данной схемы (модели);

роль коллективного решения задач в этом году существенно возрастает из-за более высокого уровня сложности самих задач. При этом не следует опасаться пассивных членов микроколлектива; необходимо лишь, чтобы каждый ученик мог самостоятельно ответить на вопросы по поводу решенной коллективом задачи;

дискуссии по поводу обсуждаемых вопросов должны служить цели выработки культуры полемики. Не следует бояться провоцировать дискуссии даже по самым «заумным» вопросам. Учитель в данном случае (как, впрочем, и всегда!) не должен выглядеть оракулом, которому заранее известны ответы на самые сложные вопро-

сы. Роль учителя заключается в *ведении дискуссии, оппонировании* «докладчикам» и в обязательном четком *подведении итогов* обсуждения: обозначении позиций, выяснении, что удалось установить совместными усилиями, а что нет. При появлении возможности эту роль следует полностью или частично поручить одному из учеников.

Следует помнить, что за три года изучения курса накоплен серьезный эмпирический и теоретический материал; таким образом, задача девятого класса — обобщение, осмысление и применение методологии к различным областям знания.

Тематическое планирование (2 ч/нед × 34 нед = 68 ч)

1. Модели и моделирование (4 ч)

Понятия *объекта моделирования и модели*. Невозможность или затруднительность исследования определенных свойств объекта заставляет обращаться к исследованию аналогичных свойств модели. При этом другие свойства объекта и модели (как и вообще природа объекта и модели) могут быть различными.

Построение и применение модели. Компьютерный эксперимент.

«Воспоминание» о моделях мышления. Эксперименты *натурные и мысленные* (общее представление). Сознание и подсознание. Как строить модели? Необходимость *построения моделей с определением границ их применимости*.

2. Диалектический метод (5 ч)

«Воспоминания» о моделях человеческого мышления: эволюционирующая база знаний («два этажа»), последовательность качественных «модификаций» баз знаний («три этажа»).

Каков он, этот «третий этаж»? В каком смысле он «вознесен над знаниями»? Что можно и чего нельзя сделать, применяя этот метод?

Диалектика — в буквальном переводе «беседа двоих». Название достаточно четко отражает сущность:

«дихотомический» характер представленный при применении диалектического метода;

«беседа» — вовлеченность в информационные процессы, т.е. бессмысленность упоминания о диалектике вне познающего человека (субъекта).

Диалектический метод предполагает рассмотрение пары определенным образом подобранных общих понятий (категорий). Не акцентируя на этом внимание, мы уже применяли диалектический метод еще в интегративном предмете «Окружающий мир», и в особенности в курсе информатики.

Вспомним, когда (начиная с 6-го класса)

понятия возникали у нас не поодиночке, а парами. Таких примеров достаточно много:

в 6-м классе:

интеллектуальное и материальное (информация и ее носитель), творческое и нетворческое, необходимое и случайное (закономерное и вероятное);

в 7-м классе:

логические связки и и или; кванторы существования и всеобщности; факты и правила; базис индукции и индуктивный переход; прямая и обратная теоремы; индукция и дедукция («снизу вверх» и «сверху вниз»);

в 8-м классе:

база знаний и алгоритм; процесс и результат (историческое и логическое).

Не всякая пара категорий будет являться диалектической! Необходимы опять же два момента:

две данные категории вместе должны представлять единое целое;

существенно (в известном смысле — диаметрально противоположно) отличаться друг от друга.

Насколько удачно выбрана категориальная пара («ракурс рассмотрения» объекта), настолько удачно будет применен метод. Один «угол зрения» породит излишне сложный (а потому и нереализуемый в данное время) процесс рассмотрения, а другой — даст возможность получить очень простую и наглядную классификацию.

3. Закон взаимоперехода количественных и качественных изменений (3 ч)

Мы уже знаем по моделям человеческого мышления, что процесс развития протекает эволюционно, накапливая в определенном направлении *количественные* изменения, а затем в результате *качественного* «скачка» изменяется (корректируется) само «направление движения».

Вспоминая взаимосвязь онто- и филогенеза в биологии, мы убеждаемся, что те же закономерности присущи и биологическому развитию. Один и тот же результат при этом может достигаться, вообще говоря, различными способами; дискретные же «ступени» процесса развития соответствуют основным структурным компонентам полученного результата.

Итак, развитие происходит в рамках одного «качества», пока количественное накопление не сделает эти рамки «тесными». При этом возникает «кризис», приводящий к скачку — смене «качества». Затем следует процесс накопления иных количественных характеристик уже в рамках нового «качества» и т.д.

Процесс развития вообще протекает таким же образом, как и уже знакомый нам процесс смены мысленных моделей: в результате накопления количественных харак-

теристик происходит «кризис», вследствие которого модель скачком меняется и т. д.

4. Закон отрицания отрицания (3 ч)

Итак, процесс развития неоднороден, помимо чисто количественного роста он содержит также «сгустки» приобретения новых качеств.

«Задача о беглеце». «Четность» и «нечетность». Двоичные деревья. «Сбалансированное» и «несбалансированное» деревья. Практикум на Прологе.

Какова же последовательность этих «качеств», возникающих в процессе развития? Есть ли какая-либо взаимосвязь между «соседними» (по порядку) качествами?

Вспомним закон «двойного отрицания», возникший в 7-м классе при рассмотрении формально-логической модели человеческого мышления.

$$A \rightarrow \text{не } A \rightarrow \text{не (не } A) = A.$$

Процесс представляет собой как бы замкнутый круг: первое отрицание порождает нечто вполне противоположное A , второе отрицание приводит в точности к отправной точке (A). Вообще при продолжении процесса все четные отрицания совпадают с A , все нечетные — с отрицанием A .

Чем отличается более совершенная модель мышления (да и развития вообще) от формально-логической? Из «плоской» модель становится «трехмерной»: при этом круг превращается в подобие спирали. Здесь второе диалектическое отрицание A — уже не само A , но нечто очень похожее на A с определенной точки зрения («расположенное на более высоком уровне»). Этот эффект возникает из-за чередования формальнологических моделей.

Таким образом, в диалектике закон двойного отрицания превращается в закон отрицания отрицания.

Проявления закона в различных предметных областях.

5. Закон единства и борьбы противоположностей (3 ч)

Возникает закономерный вопрос: что является движущей силой столь своеобразного по своим закономерностям процесса развития? Что обуславливает попеременный (непрерывно-дискретный) характер этого процесса, более того, в известном смысле, «сворачивает» его в спираль? Есть ли это *внешний* фактор по отношению к развивающемуся объекту или это сугубо *внутреннее свойство* самого этого объекта?

Устойчивая повторяемость свойств процесса развития наводит на мысль, что в основе развития лежит присущий самому объекту механизм последовательного порождения «движения» в определенном на-

правлении до наступления некоторого критического «предела», после чего на место прежнего «движителя» заступает новый и т. д.

Будучи рассмотрен под определенным «углом зрения», объект может быть представлен как «сосуществование» двух сущностей (частей), между которыми существует противоречие (может быть, даже не одно). Противоречие, являющееся в данное время ведущим, обуславливает «конфликтный» характер этого «сосуществования». Когда конфликт достигает апогея, происходит «снятие» противоречия. При этом противоречие, бывшее ведущим, перестает быть таковым и «уступает свое место» другому из имеющихся в наличии противоречий.

Поэтому при анализе явлений нужно уметь «расчленив» исследуемый объект на две такие сущности, противоречие между которыми является в данное время ведущим. После этого для выявления ведущего противоречия может потребоваться смена рассматриваемой «пары».

Таким образом, при анализе поведения интересующего нас объекта очень важно выбрать «ракурс наблюдения», чтобы выявить ведущее противоречие (с точки зрения интересующих нас свойств этого объекта). При этом чем меньше допущений требуется для описания объекта, тем выше ценность проведенного исследования. Чем реже нас ставит в тупик вопрос «почему?», тем выше научный уровень исследования.

Проявления закона в различных предметных областях.

6. Часть и целое. Их взаимодействие (6 ч)

Каков окружающий мир? Он таков, каков есть; он существует объективно, т. е. независимо от того, что мы о нем думаем, и думаем ли вообще. Объективно существуют не только материальные, но и духовные (интеллектуальные) явления.

Поскольку мы не можем познать окружающий мир целиком, мы вынуждены исследовать его лишь фрагментами. Однако выделить нужную часть можно удачно либо неудачно (пример с тортом). Чем же нужно руководствоваться, выбирая фрагмент всей объективной реальности в качестве «части» для исследования?

Прежде всего, разумеется, простотой рассмотрения интересующих нас явлений внутри этой части. Но этого недостаточно! Часто забывают о необходимости рассматривать взаимоотношение части со всем целым, т. е. со всей объективной реальностью. Это взаимодействие двунаправленное (целое влияет на часть, но и часть влияет на целое).

Таким образом, нужно выбрать разумный компромисс между простотой «части» в данном рассмотрении и простотой взаимоотношений «части» и всего «целого». За все «приходится платить»: выигрывая в одном, проигрываешь в другом. Пример: метод пошаговой детализации в алгоритмизации и программировании. Алгоритмический практик.

7. Субъект и объект. Процесс познания (6 ч)

Как мы уже неоднократно наблюдали (начиная с 6-го класса), в анализе объективно существующих явлений тем не менее очень велика роль того, кто занимается изучением этих явлений (субъекта). Все дело в том, что субъект, познающий мир, — не «удаленный наблюдатель», а активный участник происходящих (взаимосвязанных!) объективных процессов.

Поэтому процесс взаимодействия субъекта (т. е. изучающего объективную реальность) и объекта (т. е. самой объективной реальности) — процесс двусторонний. Этот процесс, называемый процессом познания, как правило, предполагает не просто «наблюдения», но и активную позицию познающего субъекта, его деятельность.

Диалектический метод позволяет попарно вычленять интересующие нас аспекты этого процесса. Процесс включает в себя определенное воздействие на объект, а затем осмысление «отклика» на это воздействие. Возникающее при этом (и обостряющееся с течением времени) противоречие требует следующего «зондирования» и т. д. Располагая методологией проведения таких экспериментов, субъект формирует и периодически обновляет свою «базу знаний».

При ненаучном способе познания этот процесс протекает бессистемно и подчас приводит к глубокому кризису; научный же способ позволяет облегчить постижение закономерностей чередования этих моделей.

Проведение мысленных экспериментов. Осмысление роли подсознания в познавательном процессе.

8. Живое и неживое. Жизни как процесс (6 ч)

Человек является частью живой природы. С глубокой древности людей интересовал вопрос об отличии живого от неживого. Попробуем применить диалектический метод и здесь.

Отметим, что традиционно противопоставляемые жизнь и смерть не являются диалектической парой (ибо жизнь — это процесс, а смерть — лишь момент прекращения этого процесса; вместе они не составляют единого целого). Диалектическая пара «живое и неживое» (или «жизнь» и «не-жизнь») отражает как структуру мира, так и опре-

деленный процесс (вспомним диалектику исторического и логического). «Существование» живого и неживого в мире можно соотносить с процессом взаимоперехода живого в неживое и наоборот (1). В этом смысле смерть представляет собой неизбежный момент перехода части живого в часть неживого для того, чтобы, замкнув цикл, вновь через некоторое время трансформироваться в живое.

С принципиальной точки зрения (рассматривая часть и целое) живые системы — это системы открытые. Открытость части как системы заключается в характере ее взаимоотношений с целым. Живые системы — это системы, поддерживающие относительную стабильность за счет целенаправленного взаимодействия с окружающей средой. При этом стабильность поддерживается в большей степени не на уровне клетки, а на уровне органа, особи, популяции, вида, рода и т. д.

9. Человек и общество. Социальные процессы (9 ч)

Человек является частью не только живой природы, но и общества себе подобных. Общество — не просто сумма отдельных составляющих его людей. Общество — это система, своеобразный «организм», каждая «клетка» которого наделена способностью к мышлению.

Можно ли говорить о том, что общество подчинено каким-либо законам? Или же оно представляет собой сборище отдельных людей, каждый из которых движим лишь собственным свободным волеизъявлением?

Достаточно значимые сходные потребности людей обуславливают поведение общности людей направленным образом. Поэтому есть смысл говорить о том, что общественное развитие подчинено вполне объективным законам.

Влияние личности на ход социальных процессов. Понятие духовности. Возможность субъективного влияния на ход процесса общественного развития.

10. «Эгоистическое» и «прагматическое» (5 ч)

Рассмотрение человека с трех точек зрения: биологической, социальной, гносеологической (познавательной).

Цели и критерии деятельности человека (человечества). Противоречие между интересами человека и человечества в целом.

11. Противоречия эпохи и пути их разрешения (10 ч)

Противоречие между «экологическим» (антитехнологическим) и «экономическим» (потребительским) императивами. Проявления противоречия. Примеры «крайностей» в решении проблемы. Ограниченность «только природоохранного» подхода к решению эко-

логических проблем. Противоречит ли научно-технический прогресс решению экологических проблем?

Производство и потребление. Материальные и духовные ценности; баланс их потребления. Естественное и гуманитарное знание; искусство в жизни общества.

Резерв времени (за год) — 8 ч.

Инварианты успешного завершения курса 9-го класса

К концу обучения в 9-м классе учащиеся должны обладать следующими представлениями, закрепленными в практической деятельности (по разделам).

1. Методология моделирования.

Процесс моделирования включает следующие моменты. Предположим, что в некоторый момент нас интересуют определенные свойства некоторого реально существующего объекта (собственно, в этом и заключается всякая задача). Не имея возможности непосредственно исследовать интересующие нас свойства объекта, мы конструируем другой объект, быть может, совершенно иной природы), который объединяет с исходным объектом одинаковое проявление свойств, которые мы хотим исследовать, у обоих этих объектов. Второй объект мы называем моделью первого и приступаем к изучению интересующих нас свойств второго (а значит, и исходного).

Процесс моделирования включает в себя процесс построения модели и ее исследования. Выбираемый нами метод познания («третий этаж» мышления) призван помочь, преодолевая возникающие в процессе решения задач кризисы, выходить из них, выстраивая следующую модель. После того как наши потребности приведут к «выходу за границы» и этой модели, с помощью метода познания мы должны построить следующую модель и т. д.

В качестве метода познания мы используем диалектику. Законы диалектики; эволюция от формальной к диалектической логике. Формально-логическое и диалектическое противоречия.

Парные понятия и категории диалектики. Необходимое и случайное, историческое и логическое. Рассмотрение диалектических пар двойным образом: как структуры состояния и как структуры процесса. Часть и целое. Субъект-объектное отношение. Живое и неживое.

2. Диалектический практикум.

Решение (в том числе коллективное) актуальных задач из реальной практики взаимодействия человека и природы, человека и

общества, субъекта и объекта познания. Системный анализ.

Решение задач посредством: введения пары понятий, установления их единства и противоположности (выявление противоречий), распространения их как «крайних точек» с введением меры между ними, рассмотрения взаимодействия и «взаимопроникновения» понятий как структуры и как процесса.

3. Творчество с точки зрения взаимодействия человека и окружающей среды.

Три сущности человека: биологическое существо, социальное существо, субъект процесса познания. Диалектическое обоснование данной классификации.

Цели и критерии деятельности человека и человечества. Объективное противоречие между интересами человека и человечества в целом. Пути снятия этого противоречия.

Противоречия между человеком (человечеством) и окружающей средой (во всем ее многообразии). Экологические проблемы второй половины XX века.

Глобальные противоречия эпохи. Противоречие между «экологическим» (понимаемым как антитехнологический) и «экономическим» (понимаемым как потребительский) императивами. Ограниченность «крайностей» в попытке снятия этого противоречия. Возможные пути решения проблемы. Необходимость привлечения всей мощи научно-технического прогресса не во имя увеличения «стандарта потребления», а напротив, капиталоемкие и наукоемкие вложения как в ориентированные на потребление, так и в природоохранные мероприятия.

Рекомендуемая литература

По 9-му классу

1. Ильенков Э. В. Об идолах и идеалах.— М.: Политиздат, 1968.
2. Лившиц М. А. На деревню дедушке (в мире эстетики).— М.: НИИ теории и истории изобразительных искусств СССР, 1990.
3. Мичи Д., Джонсон Р. Компьютер-творец.— М.: Мир, 1987.
4. Пуанкаре А. О науке.— М.: Наука, 1990.
5. Тарасов Л. В. Современная физика в средней школе.— М.: Литпросвещение, 1990.
6. Щеголев А. Г. Все как взаправду (Кризис образования и регионы опережающего развития) // Развитие.— 1992.— № 2—3.

Программа курса «Основы информатики и вычислительной техники»

Примерное тематическое планирование

VIII класс

1-я четверть (9 ч)

Понятие об информации. Виды информации. Способы поиска, обработки, хранения информации. (1 ч). Общий вид ЭВМ. Составные части компьютера. Их назначение и взаимосвязь. Назначение локальной сети. (1 ч). Что изучает информатика. Информатика как наука. Определение информатики. Роль информатики и вычислительной техники на современном этапе развития общества. Правила техники безопасности при работе на ЭВМ. (2 ч). Игровая программа. Знакомство с клавиатурой ЭВМ. Работа с программой «Клавиатурный тренажер». (2 ч). Работа с графическим редактором. Операции рисования. (2 ч). Самостоятельная работа (1 ч).

2-я четверть (7 ч)

Работа с текстовым редактором. Редактирование текста. Вставка/замена символов. Вставка/замена строк. Форматирование строк, абзацев, страниц. Копирование, перенос, замена слов, строк, абзацев. Удаление символов, строк, абзацев. (4 ч). Самостоятельная работа (1 ч). Работа с музыкальным редактором. Нотная запись. Редактирование и исполнение мелодий. (2 ч).

3-я четверть (10 ч)

Работа с ЭВМ в режиме непосредственных вычислений. Работа с электронными таблицами. (3 ч). Самостоятельная работа. (1 ч). Работа с базами данных. Понятие таблицы. Заполнение и редактирование таблиц. Поиск информации в таблицах. (2 ч). Запись и поиск информации в базах данных. Создание новых таблиц. Работа по сохранению и введению информации в базы данных. (3 ч). Самостоятельная работа. (1 ч).

4-я четверть (8 ч)

ЭВМ — электронное устройство обработки информации. Виды и свойства информации. (1 ч). Системы счисления. Кодирование информации. Двоичное кодирование. (3 ч). Единицы измерения объема информации. Архитектура ЭВМ. (1 ч). Средства длительного хранения информации. Понятие файла. Операции записи, считывания на диск, ленту при работе с графическим, текстовым редакторами, базами данных. Первоначальные сведения об операционной системе, ее назначение. (3 ч).

Функциональная организация ЭВМ. Организационная структура оперативной памяти. (1 ч). Понятие ячейки памяти, ее свойства. Принцип программного управления. Принцип хранимой программы и данных. Принцип адресности. (3 ч). Формы представления целых, вещественных, символьных данных в ЭВМ. Классификация программных средств. (2 ч). Контрольная работа. (1 ч). Введение в логику. Суждения. Рассуждения. Элементы формальной логики: высказывания, составные высказывания, логические связи (2 ч). Таблицы истинности. Математическая логика. Понятие конъюнкция, дизъюнкция, отрицание, импликация. Составление логических выражений. (2 ч). Основные законы математической логики. (2 ч). Составление и упрощение логических выражений на основе законов математической логики. Использование математической логики в технике. Элементы контактных и полупроводниковых схем. Упрощение логических выражений на основе законов логики. Реализация функций математической логики на Прологе. Тестирование полупроводниковых и контактных схем на Прологе. (4 ч). Контрольная работа. (1 ч).

IX класс

1-я четверть (18 ч)

Работа на Прологе. Понятие факта, вопроса, правила. Составление базы знаний. (2 ч). Правила логического вывода. Поиск в базе знаний. Понятие переменной (параметра). (6 ч). Графические средства Пролога. Организация рекурсий. Введение параметра-счетчика. (4 ч). Рекурсивное задание числовых функций (1 ч). Самостоятельная работа. (1 ч).

2-я четверть (14 ч)

* Окончание. Начало см.: Информатика и образование. — 1993. — № 1.

3-я четверть (20 ч)

Понятие алгоритма. Знакомство с исполнителем алгоритма. Система команд исполнителя. Свойства алгоритмов. (2 ч). Алгоритмический язык как средство представления записи алгоритмов. Правила записи алгоритмов. Ошибки составления алгоритмов. Последовательное выполнение команд. Команды ветвления, повторения. (12 ч). Вспомогательные алгоритмы. (4 ч). Самостоятельная работа. (2 ч).

4-я четверть (16 ч)

Алгоритмы обработки данных. Понятие переменной (имя, значение). Типы переменных: числовые, текстовые, табличные. (1 ч). Действия над данными. Отношения между данными в качестве условий. (4 ч). Аргументы и результаты алгоритмов. (1 ч). Условие задачи, математические способы ее решения. Построение алгоритмов. (7 ч). Самостоятельная работа. (1 ч). Понятие о программировании. Профессиональная специализация, связанная с обслуживанием ЭВМ, ее программным обеспечением, подготовкой данных. Использование ЭВМ в науке, народном хозяйстве, быту. (2 ч).

X класс

1-е полугодие (32 ч)

Повторение алгоритмических структур. (4 ч). Повторение понятия файла. Основные команды операционной системы. Вывод каталога. Запись и считывание файла. Удаление файла. Запуск программы. Назначение основных функциональных клавиш. (2 ч). Этапы обработки программ. Понятия об интерпретаторах и компиляторах. Сообщения об ошибках (на примере готовой программы). (2 ч). Начала программирования. Алфавит языка программирования. Типы переменных. Арифметические операции. (2 ч). Операция присваивания. (1 ч). Логические операции. (1 ч). Библиотека встроенных функций. (1 ч). Редактирование программ. Классификация ошибок. (1 ч). Разработка алгоритмов и составление программ по этапам:

- а) линейные алгоритмы, операция ввода-вывода информации (2 ч);
- б) разветвляющиеся алгоритмы (3 ч);
- в) алгоритмы, объединяющие линейную и разветвляющуюся структуры (3 ч);
- г) циклические алгоритмы (4 ч);
- д) алгоритмы, объединяющие линейную, разветвляющуюся и циклическую структуры (4 ч).

Решение задач выполняется по схеме: математическая постановка задачи; выбор способа решения;

разработка сценария;
разработка алгоритма;
анализ правильности алгоритма;
кодирование алгоритма на язык программирования;
составление плана ввода и отладки программы;
анализ результатов.
Контрольная работа (1 ч).

2-е полугодие (36 ч)

Работа с таблицами (2 ч). Одномерные и двумерные массивы (22 ч). Вспомогательные алгоритмы (8 ч). Этапы решения задач на ЭВМ:

постановка задачи;
построение математической модели;
выбор методов решения;
разработка сценария;
разработка алгоритма с использованием метода пошаговой детализации;
анализ правильности алгоритма;
разработка тестов испытания программы;
кодирование программы на язык программирования;
составление плана ввода и отладки программ;
анализ конечных и промежуточных результатов.

Отработка этапов решения выполняется на задачах из предметов школьного курса и задачах по интересам. Контрольная работа (2 ч).

XI класс

1-е полугодие (16 ч)

Повторение команд операционной системы по вводу-выводу информации на магнитные носители. Организация файлов. (2 ч). Понятие вычислительного эксперимента. (1 ч). Примеры постановки и проведения вычислительного эксперимента:
для физических процессов;
в биохимических задачах;
для решения экологических и экономических проблем.

Создание математической модели в задачах вычислительного эксперимента с использованием несложных численных методов (приближенное вычисление площадей, решения уравнений методом половинного деления, приближенного вычисления значений функций и других). (11 ч). Контрольная работа. (2 ч).

2-е полугодие (18 ч)

Краткая история развития вычислительной техники. Представления о развитии элементной базы. (2 ч). Поколения ЭВМ.

Их принципиальные различия. Развитие программного обеспечения. (1 ч). Дополнительные сведения об устройстве ЭВМ. Использование вычислительной техники в народном хозяйстве, быту, науке. (2 ч). Работа над индивидуальным заданием. Отчет о выпол-

ненной работе (выполняется в течение полугодия).

Программа разработана на основании предложений Координационного Совета заведующих УМК информатики и ВТ России.

КОМПЬЮТЕРНЫЕ ИГРЫ (КИ-ЧПМ)

предлагают:

Программное обеспечение для пользователей компьютеров БК-0010, 0010.01, БК-0011М, ДВК-2М, 3М, УКНЦ (МС-0511), ZX-Spectrum, ХОББИТ ИК8030(128К), ПОИСК, МК-88, IBM PC, классов информатики КУВТ-86, КУВТ-УКНЦ.

Кооператив располагает обширным банком игровых, учебных, системных, прикладных программ.

!!! РАСЦЕНКИ ВДВОЕ, ВТРОЕ НИЖЕ РЫНОЧНЫХ !!!

Списки программ высылаем бесплатно всем желающим, но прежде всего тем, кто:

- * *обозначит тип своего компьютера прямо на конверте;*
- * *вложит в письмо конверт с наклеенными марками и надписанным обратным адресом.*

Для пользователей БК-0010, 0010.01:

- * *турбокопировщик, многократно повышающий плотность и надежность записи - бесплатно, при условии заказа программ;*
- * *ПЗУ с разнообразными программами.*

Переписка: 189510, Санкт-Петербург, Ломоносов, а/я 649.

Множество недавних статей в компьютерных журналах больше похожи на инструкции по нажиманию на клавиши компьютера, которые формируют у читателя так называемое «кнопочное мышление», но к сожалению, не прибавляют новых идей. Статья профессора К.Видерхольда и его коллег выгодно отличается от указанных выше работ. Она посвящена одной из актуальнейших тем современного образования: использованию компьютеров в начальной школе. Наши немецкие коллеги неторопливо и глубоко рассматривают положение дел в этой области как в Германии, так и за рубежом. Нам особенно близок подход коллег из Германии, их ориентация на практическое использование компьютеров в начальной школе, ориентация на творческую работу и на получение детьми конкретных результатов. Они прислали нам журналы творческих работ (изданные самими учащимися) со стихами и рисунками, видеofilm с методикой обучения в 4 классе.

Начавшаяся год назад реализация проектов «Пилотные школы» и «Пилотный детский сад» на базе компьютеров IBM PS/2 позволяет реализовать разработанную концепцию непрерывного компьютерного образования на более комфортабельной технике, которой уже давно обладают учителя школ за рубежом. И не беда, что наши коллеги из Германии работают на компьютерах Эппл и Макинтош, а мы на Ямахах и IBM. Не менее важен обмен глубокими и плодотворными идеями и методами обучения, чем конкретными компьютерными программами. Идея Сеймура Пейперта об управлении черепашкой с помощью компьютера появилась более двадцати лет назад. С тех пор сделано множество других компьютерных исполнителей, но не появилось такого же множества плодотворных идей.

Я.Марголис

А.ВИДЕРХОЛЬД, Х.МИТЦЛАФФ, У.ГРИС

23

Компьютер в начальной школе

Вопрос об использовании компьютера на занятиях в начальной школе вызывает сомнения у многих практиков и теоретиков начального образования в Германии, а некоторые даже категорически отвергают такую возможность. Однако, компьютеры реально используются во многих семьях, что заставляет серьезно рассматривать эту проблему.

Отец одного из учеников начальной школы, Даниеля, за которым мы наблюдали, купил для работы портативный персональный компьютер. Чтобы быстрее его освоить, он приносил компьютер на выходные дни домой, и Даниель, которому в то время было восемь лет, тоже познакомился с компьютером. Как и большинство детей в этом возрасте, Даниелю захотелось подражать отцу. Тот, в свою очередь, поддержал проявленный сыном интерес и постепенно стал объяснять основные понятия, связанные с компьютером, а также работу некоторых программ. Скоро у Даниеля появилась собственная дискета с созданными текстами и рисунками, а затем и целый архив.

Через два года Даниель без труда постоянно выдумывал небольшие текстовые и рисованные истории, в которых стремился отразить собственные переживания, а также искусно оформлял разными значками приглашения, поздравления, загадки. Он работал над небольшой классной газетой для своих

друзей и для него уже не было незнакомым понятие «настольное издательство».

После наблюдений за детьми в домашних условиях мы почувствовали необходимость задуматься над темой «Компьютер в начальной школе» в педагогическом аспекте.

В настоящее время в Германии выпускается все большее число учебных программ для начальной школы. Многие из них используются родителями для учебных целей. Однако систематический педагогически ориентированный анализ учебного программного обеспечения, который мог бы оказать помощь родителям, находится сейчас лишь в стадии становления.

Детальные и научно-обоснованные исследования, позволившие бы оценить возможности и недостатки использования компьютеров в начальной школе пока в Германии не проводились. В Зиммермане был проведен эксперимент, при котором начальная ступень школы сознательно была включена в концепцию информационно-технического начального образования. В Зиммермане использовали систему TOAM, разработанную в Израиле, которая содержит 32 рабочих места с мониторами. Дети занимались 1-2 раза в неделю по 20 мин по математике и языку с использованием тренировочно-контролирующей программы, которая указывала на недостатки в знаниях по ранее изученным темам.

Кроме этого проекта, нам известно всего лишь несколько попыток, когда учителя начальных школ использовали в работе личные бытовые или профессиональные персональные компьютеры.

Согласно ряду американских публикаций, все программы учебного назначения можно разделить на следующие группы:

- учебно-тренировочные программы для развития и закрепления основных знаний и навыков (умений);
- программы для введения нового учебного материала и контроля усвоения (Тьютор);
- имитационные программы;
- программы обучения языкам программирования;
- прикладное программное обеспечение (для редактирования текстов или графики, верстки, управления базами данных и т.д.);
- игровые программы.

В то время как в англосаксонских странах имеются специально разработанные для начальной школы программы, в Германии в настоящее время хорошо разработаны только программные продукты первой группы.

Но совершенно очевидно, что для разработки нового высококачественного программного обеспечения совершенно необходимо тесное сотрудничество квалифицированных специалистов по программному обеспечению и опытных педагогов начальных классов. При этом именно для самых младших школьников следует требовать создания самого лучшего, дидактически продуманного и мотивированного программного обеспечения.

Контролируемые эксперименты по использованию компьютеров на уроках в начальной школе должны учитывать цели и принципы обучения в начальной школе, которые в последние годы все в большей степени ориентируются на прогрессивные мировые тенденции.

Информационные и воспитательные цели модельного эксперимента по использованию компьютера на занятиях в начальной школе можно представить следующим образом:

- учащиеся должны уметь критически подходить к новым информационным технологиям (НИТ), не пренебрегая непосредственно контактом с людьми;
- наряду с ознакомлением с техническими, экономическими и социальными аспектами НИТ, ученики должны знать о пределах из возмож-

ностей и опасностях, связанных с их применением;

- НИТ следует вводить на уроках преимущественно в тех случаях, когда они представляют ученикам новые возможности для творческого самовыражения;
- на уроке должны затрагиваться, перерабатываться и при необходимости корректироваться сведения, полученные детьми в дошкольном возрасте и вне школы;
- школьникам должны быть предоставлены возможности раскрытия и анализа особенностей и связей между конкретными, схематическими и символическими знаниями.

При этом необходимо ответить на ряд вопросов.

- Имеются ли компьютерные программы по математике и немецкому языку, способствующие лучшей обучаемости младших школьников арифметике, чтению и правописанию? Какова структура этих программ?
- Имеются ли программы, поддерживающие такой учебный процесс, который без компьютера был бы затруднен или вообще невозможен?
- Имеются ли рациональные с педагогической точки зрения учебные программы, совместимые с учебным планом?
- Имеются ли подходящие программы для занятий по музыке, способствующие лучшему пониманию музыки и развитию навыков работы с нотами? Как построены такие программы?

Мы специально выделяем занятия по искусству, в которых могут быть использованы различные графические программы, так как применение компьютеров в этой области вызвало бы недопустимое ограничение детской непосредственности и творческого начала: современные программные продукты значительно уступают творческим возможностям детей.

Возникает и ряд вопросов более общего характера.

- Можно ли уже на данном этапе указать пределы возможностей применения компьютеров?
- Какое влияние окажет использование компьютеров на организационные формы занятий в начальной школе? Будет ли это стимулировать групповую работу или будет способствовать размежеванию учеников?
- Будет ли применение компьютеров способствовать дифференцированию форм проведения занятий и их большей открытости?
- Приведет ли работа с компьютером к рас-

крытию новых, ранее скрытых творческих возможностей?

- Будет ли способствовать использование компьютеров развитию самостоятельности учеников?
- Как подействует компьютер на мотивацию учебы детей?
- Как реагируют учителя и родители на использование компьютера в начальной школе?
- Годится ли компьютер в качестве инструмента для подготовки и издания документации, создания единого школьного банка данных и, наконец, для творческой работы?

Остановимся на использовании компьютера как инструмента.

В одной из начальных школ проводилась целевая учебная неделя. Было организовано 20 групп. Учебная группа, состоящая из учеников 3 и 4 классов, получила задание: выпустить под руководством учительницы газету, посвященную проведению этой недели.

На первом уроке учительница рассказывала о традиционном способе типографского набора. Затем ребята знакомилась с современной техникой, причем работа с **ТЕКСТОВЫМ РЕДАКТОРОМ** многим кажется **БОЛЕЕ ЛЕГКОЙ** и удобной, чем **НАБОР ИЗ СВИНЦОВЫХ БУКВ** (традиционный типографский набор).

В начале недели ребята посетили редакцию местной газеты, беседовали с журналистами, осматривали редакционные помещения, типографию. Затем была принята следующая последовательность работ по выпуску собственной газеты:

- наблюдение за работой редакции;
- описание работы и распределение заданий по группам;
- подготовка статей и их обсуждение;
- компоновка отдельных страниц, включая размещение иллюстраций и текста, подбор заголовков и т.д.;
- распечатка пробного варианта, в котором содержатся материалы, подготовленные всеми 20-ю группами;
- просмотр пробного варианта, его обсуждение, правка и корректировка верстки;
- распечатка и тиражирование окончательной версии;
- распространение газеты среди учеников.

Приблизительно так, с элементами игры школьники учатся пользоваться компьютером как хорошим инструментом: легко и

быстро исправляются ошибки при печати, за короткое время можно составить обширный текст, менять число строк, интервал между ними, вид шрифта и т.д.

Аналогично можно организовать эксперименты и с другими видами изданий.

Сеймур Пейперт ответил на вопрос, требуются ли младшим школьникам языки программирования, в своей известной книге «Озарения» однозначно: «Да». Специально для детей он придумал язык ЛОГО. Тем не менее, чтобы получить обоснованный ответ на поставленный вопрос, необходимы эксперименты. Учитывая современный уровень развития вычислительной техники, мы склонны к отказу от обучения элементарному программированию в младшем школьном возрасте. Главная мысль в работе С.Пейперта - дать ребенку возможность управлять компьютером, а не превращать его (ребенка) в подручного машины. Сегодня благодаря новой технике этот замысел можно реализовать более простым методом. С помощью «мыши» и меню младшие школьники могут творчески работать со сложными графическими программами, текстовыми редакторами и программами компьютерной верстки, не имея даже навыков программирования. Благодаря этим новым технологиям, которые, естественно, будут совершенствоваться, специальные знания по ЭВМ (различные языки программирования, методы разработки программ и т.д.) будут требоваться лишь от ограниченного круга специалистов. Учитывая такие прогнозы, зачем принуждать всех детей к изучению подобных специальных вопросов?

В заключение стоит еще раз подчеркнуть: в будущих экспериментах не следует делать упор на введение различных технических «новинок», превращающих начальную школу в открытый рынок для новых технологий. Педагогика начальной школы призвана сегодня критично подходить к меняющейся действительности, окружающей детей в высокотехнологизированном мире. При этом следует объективно изучать и сравнивать как негативные воздействия, так и позитивные дидактические возможности новых технологий, чтобы выработать ориентированную на будущее методику и практику построения учебных занятий в начальной школе.



Министерство образования Российской Федерации

Центр Интерактивных Средств Обучения

125315, г. Москва, ул. Часовая, д. 21-6

телефон: 155.87.30., 155.87.37. факс: 155.87.27,

Audiovision in Perfektion

Официальный представитель в России
фирмы **TELEMEDIA OMBH**
концерна **BERTELSMANN AG**



Бизнес-партнер фирмы

IBM

ПРЕДЛАГАЕТ УЧЕБНЫМ ЗАВЕДЕНИЯМ

- Широкий выбор программно-методических комплексов по различным предметам школьного курса для IBM-совместимой техники, снабженных методическими и дидактическими материалами.
- Инструментально-педагогические средства для изготовления контрольно-обучающих программ, систем моделирования, составления расписания, тарификации учителей, контроля успеваемости.
- Видео-программы для вузов, школ, колледжей.
- Проведение обучения Ваших специалистов.
- курс информатики для УКНЦ (Roo & Roby и Schoolworks), известные по проекту "Пилотные школы".
- а также средства технической поддержки:

Интерактивная аудио-визуальная обучающая система (производство фирмы **TELEMEDIA**). Система представляет собой комбинацию из персонального компьютера и проигрывателя для оптических видеодисков.

Видеосистема "TV-NET" - это телевизионная кабельная сеть для трансляции программ на несколько телевизоров одновременно.

Новые компьютерные классы фирмы IBM на базе персональных компьютеров PS/1 Model 2121-642.

Локальные вычислительные сети семейства VOXnet (VOXnet/2, EasyVOX).

Сети VOXnet объединят ваши персональные компьютеры в информационную систему.

СКИДКА

- для оптовых покупателей
- для школ-участников "Пилотного проекта".

Желающие могут получить каталог аудио-визуальных и компьютерных средств обучения, распространяемых центром.

Приглашаем работать с нами на дилерских условиях.

Д. БОГДАНОВА, А. ФЕДОСЕЕВ, С. ХРИСТОЧЕВСКИЙ

Телекоммуникации для образования

Практическое использование электронной почты в наших школах (в основном, московских) началось в 1989 г. Сегодня можно выделить различные подходы к организации систем электронной почты.

Наиболее простой и естественный способ — установление связи с группой школ за рубежом, объединенных каким-либо центром. Такая система характеризуется наличием устойчивого канала связи, к которому с каждой стороны подключается некоторое количество школ. Они объединяются либо в рамках учебного проекта, либо создают и реализуют проекты для пар школ.

Заслуживает внимания использование наиболее распространенных систем электронной почты (RELCOM, ADONIS) для создания на их основе некоторой образовательной подсистемы. В этом случае сохраняются все достоинства наличия специальной системы и отсутствуют проблемы ее создания. К сожалению, в настоящее время не приходится говорить о серьезных попытках реализации этого способа.

Возможна непосредственная связь «модем-модем» без использования какой-либо системы электронной почты. Такая связь дает возможность школам реализовывать двусторонние проекты.

Авторы данной статьи непосредственно участвуют в создании и функционировании системы подключения школ к крупнейшей международной образовательной системе CAMPUS 2000 (более 10 000 абонентов во всем мире), о чем и пойдет речь ниже.

В 1990 г., благодаря помощи образовательных властей графства Стаффордшир (Великобритания) появилась возможность установить связь для шести школ из Москвы и г. Переяславля-Залесского с шестью шко-

лами Стаффордшира. Было получено шесть электронных почтовых ящиков в CAMPUS 2000, шесть модемов и обещание английской стороны оплачивать связь в течение года.

Прежде всего, нам пришлось создать центр, который организует проектную деятельность на основе электронной почты в школах, обучает преподавателей, координирует всю переписку и т. п. Большим подспорьем для нас оказались материалы Национального Совета Образовательных технологий Великобритании и Объединенного министерства образования скандинавских стран. Школы связывались с зарубежными партнерами через этот центр, используя единый ящик электронной почты. Позднее, когда у школ накопился достаточный опыт, они получили модемы и стали вести переписку с центром посредством национальной телекоммуникационной системы «Русский экспресс».

На следующем этапе было принято решение о создании собственных образовательных проектов. Для этого на основе скандинавских методик и отечественных исследований был разработан метод создания образовательных проектов и апробирован на практике девятидневный курс освоения этого метода учителями. Его методологической основой является широко известный в педагогике метод проектов, позволяющий значительно повышать мотивацию учащихся и, как следствие, существенно влиять на эффективность обучения. Следует отметить характерную особенность метода, заключающуюся в том, что учителям не предлагается никаких заготовок и тем будущих проектов. Выдвижение идей и разработка

их до превращения в высокой степени готовности образовательные проекты осуществляется в процессе обучения в соответствии с методикой. В результате школам была обеспечена возможность участия в проектах, распространяемых в системе CAMPUS 2000, и развития своих собственных проектов. Что же школы получили реально?

Один из наиболее интересных проектов 1991/1992 учебного года — Worldtour (кругосветное путешествие) оказался чрезвычайно интересным приключением для наших детей, предоставивший им замечательную возможность обогатить свои знания о других странах, завести новых друзей и попрактиковаться в английском языке. Помимо этого, проект помог им больше узнать о городе, в котором они живут. Смысл проекта заключался в том, что школьники выполняли роли путешественников. Они решали проблемы виз, транспорта, обмена денег, учитывали погодные и другие условия в стране пребывания, играли роль принимающей стороны. Для этого описывались достопримечательности, составлялась программа приема гостей, создавались описания экскурсий. Московская неделя проекта была настоящим праздником для обеих участвующих школ (СШ № 141 и СШ № 159), которые работали по очереди. Дети получили два факса с Шетлендских островов, и какво же было их удивление, когда они увидели, что одно из писем написано по-русски! Дети получили открытки с видами Шетлендских островов и Австралии.

Большим успехом пользовался проект «Планета-икс», так как имел хорошую драматургическую основу. Была придумана фантастическая планета и нарисована карта одного из ее материков. Каждая группа детей могла «поселиться» на участке планеты размером 500×500 км². Причем, на первом этапе никто не должен был знать, кто является его соседом. Дети придумывали «легенды» о причинах, побудивших их лететь на эту планету, сочиняли себе новые имена и биографии, выдумывали космические корабли. Позже, высадившись на своем участке, дети придумали топографические особенности, флору и фауну. Ведущие проекта присылали сообщения о терпящих бедствие космических кораблях, о незнакомцах, желающих поселиться на планете. По всем этим проблемам ребята принимали ответственные решения. Надо сказать, что именно наши учащиеся привнесли в проект буйную фантазию и заслужили всеобщее одобрение.

Наши дети не могли в полной мере участвовать в периодически проводимом конкурсе школьных газет, поскольку в наших школах еще не получили развития настоль-

ные типографии и множительная техника. Однако они передавали материалы и заметки для других школьных газет.

Кроме этого, наши школы участвовали в нескольких «Евро-проектах» («Школа», «Окружающая среда», «Деньги»). Эти проекты интересны для наших учеников, потому что они дают представление о нестандартных темах: «Каким образом дети зарабатывают и тратят деньги?», «Что они думают об образе хорошего учителя?», «Какие качества делают учителя хорошим?». Одна из наших школ была лидером в проекте «Окружающая среда», и ученики приобрели новых друзей в Англии, Финляндии, США, Новой Зеландии.

В некоторых школах в переписке по электронной почте участвуют школьники начальных классов. В этих случаях используются тематически более простые проекты. Одна из начальных школ Стаффордшира стала партнером школы г. Переяславля-Залесского. Имея связи с Новой Зеландией, стаффордширская школа предложила своим партнерам связаться напрямую и образовать своеобразный треугольник. Обязательным сопутствующим элементом стала обычная почта, через которую передавались дополнительные материалы.

В другом случае связь с английской школой, в которой ученики изучают русский язык, стала возможной благодаря специальному программному обеспечению, предложенному английскими коллегами. Ребята писали друг другу письма, используя как английский, так и русский языки. Дети работали по совместному проекту, обсуждали стиль и уровень жизни в России и Англии.

Из сказанного очевидно, что проекты характеризуются крайним разнообразием. Во всех случаях отмечено значительное повышение интереса детей к английскому языку и информатике. Каждый проект интегрирует знания по нескольким предметам, и учителя отмечают их несомненную педагогическую эффективность.

В Великобритании электронная почта не только давно используется в учебном процессе, но с прошлого года стала обязательным элементом учебных программ.

Взаимодействие с CAMPUS 2000 дает возможность интенсивного накопления опыта работы с образовательными телекоммуникационными системами;

быстрое и эффективное подключение к системе любого количества школ (эту возможность следует рассматривать как наиболее быстрый и эффективный способ создания национальной образовательной системы электронной почты);

мгновенное включение школ в интересную проектную учебную деятельность, причем это,

как правило, интегрированные проекты, делающие возможным вовлечение учителей-предметников;

возможность реализации собственных идей и проектов;

резкое повышение интереса школьников к изучению ряда предметов, в особенности английского языка и информатики.

Накопленный к настоящему времени опыт позволяет нам реализовать следующие цели:

предоставление возможности для большинства числа российских детей непосредственного восприятия мировой культуры, осознания себя частью мирового сообщества;

развитие у детей навыков совместной (на равных) работы над коллективными международными проектами, реализуемыми с помощью новых информационных технологий;

интегрирование современных средств телекоммуникаций в школьное образование.

Мы полагаем, что достижение этих целей обеспечивается решением следующих задач:

подсоединение российских школ к всемирной образовательной телекоммуникационной сети CAMPUS 2000, обеспечивающей доступ к всевозможной информации для школьных учебных программ, междуна-

родных проектов, а также к информации для учителей и административного персонала в области школьного образования, и к другим системам;

обеспечение школ информацией о проводящихся в системе международных проектах; организация обучения и обмена опытом учителей, ответственных за ведение переписки и участие в международных проектах; привлечение творческих преподавательских сил, способных создавать содержательные проекты (в том числе телекоммуникационные), интегрированные и школьные программы. Вооружить их опытом проектной деятельности в системе CAMPUS 2000 и методом создания образовательных проектов;

своей активностью, участием в международных проектах, генерацией новых проектов сделать российскую компоненту необходимой и равноправной составной частью всемирной телекоммуникационной сети.

В свою очередь, решение этих задач зависит от финансовой помощи как федеральных, так и местных властей, а также от возможности спонсоров, поддерживающих мероприятия для детей.

29

Основы экологического земледелия и маркетинга личного и малого фермерского хозяйства

В Московском педагогическом университете (факультет «Химия и обслуживающий труд») ведутся работы по выработке и реализации концепции подготовки учителей для районных школ.

Суть этой концепции заключается в подготовке учителей, которые должны овладеть на высоком научно-методическом уровне одной основной базовой специальностью с использованием информационных технологий в обучении — в нашем случае химией и методами информатики в преподавании химии; дополнительные и трудовые специальности студент выбирает по своему желанию.

Учебная база факультета и опыт подготовки учителей по специальности «Химия и обслуживающий труд» в течение последних пяти лет позволяют предложить в качестве дополнительных специальностей:

кулинария и переработка продуктов питания;

конструирование и моделирование одежды;

информатика и вычислительная техника в обучении;

прикладное искусство и художественные промыслы.

По всем предлагаемым дисциплинам факультет располагает оснащенными современным оборудованием учебными и научно-исследовательскими лабораториями, компьютерным классом и производственно-технологической базой.

В 1993/94 учебном году на факультете состоится первый набор (в том числе и на хоздоговорной основе) студентов для получения дополнительной специальности на спецкурс: «Основы экологического земледелия и маркетинга личного и малого фермерского хозяйства» (авторы: к. т. н. доцент Недошивин В. П., д. хим. н. профессор Мансуров Г. Н.).

Этот спецкурс вводится в учебный план Московского педагогического университета по специальности 2122 «Химия и обслуживающий труд» с целью вовлечения преподавателя сельской средней школы и обучаемых им школьников в процесс постановки и организации проектов для юношества по маркетингу, экологии и хозяйствованию на селе.

Спецкурс предполагает обеспечение будущего учителя средней школы необходимым минимумом знаний, практических навыков и

методическими пособиями для самостоятельного планирования содержания школьного курса по экологическому образованию, культуре землепользования, основам планирования и ведения бюджета мелкого товарного производителя.

Спецкурс базируется на освоенной студентами системе знаний по химии, биохимии, биологии, химической технологии, технологии приготовления пищи, анализу пищевых продуктов и информатике.

Спецкурс включает цикл лекций, практических и лабораторных занятий, тренинг, ролевые развивающие игры и полевую практику. Аналогичное содержание методики рекомендуется для преподавания предмета в сельской школе в целях развития у школьников профессиональной ориентации, самостоятельности и свободы выбора, осмысленного активного подключения к хозяйственной деятельности в семье и оценке рентабельности личного хозяйства.

Чтение лекций по спецкурсу а также проведение практических и лабораторных занятий и полевой практики предусмотрено в

IX—X семестрах. К чтению лекций будут привлечены специалисты из ВНИИ Удобрений и агропочвоведения Российской академии кадрового обеспечения АПК, Международного эколого-политологического университета, российско-американского журнала «Новый фермер», Школы фермеров им. А. В. Чайнова при Московской сельскохозяйственной академии им. К. А. Тимирязева и Бизнес-центра «АГРОКОНСАЛТ».

Занятия по тренингу и развивающим играм будут проводиться в компьютерном классе кафедры химии Московского педагогического университета.

Полевая практика будет организована на Крюковской агробиостанции биолого-химического факультета Московского педагогического университета.

Предложения и заявки в произвольной форме просим направлять по адресу: 107005, г. Москва, ул. Радио, 10а, МПУ, факультет «Химия и обслуживающий труд». Телефон: 265-41-53, 261-73-94.

30

**ТЕРМИНАЛЬНЫЕ КЛАССЫ
КОРВЕТ, УКНЦ, КУВТ-86, БК0011, БК0011М
КОНСУЛЬТАЦИИ В ОБЛАСТИ ИНФОРМАТИЗАЦИИ ОБРАЗОВАНИЯ,
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ДЛЯ ТЕРМИНАЛЬНЫХ КЛАССОВ, МЕТОДИЧЕСКИЕ РАЗРАБОТКИ.**

НАУЧНО-ТЕХНИЧЕСКОЕ ПРЕДПРИЯТИЕ "АЛЬТЕРНАТИВА"



АЛЬТЕРНАТИВА

- системы программирования на языках высокого уровня;
- комплексные системно-прикладные пакеты программ;
- инструментальные системы гипермедиа;
- сетевые операционные системы;
- обучающие программы;
- мы поможем вам приобрести терминальные классы по самым низким ценам;
- дополнительная информация и рекламные материалы;

**РОССИЯ, 656049, Г.БАРНАУЛ, А/Я 3475.
ТЕЛ. (385-2) 221-050, ФАКС 236-757**

ИНФОРМАЦИОННОЕ ПИСЬМО

Министерство образования Российской Федерации, лаборатория информатизации образования Пермской области, Главное управление образования, педагогический институт, институт повышения квалификации работников образования проводят Республиканскую научно-практическую конференцию «Региональные проблемы информатизации образования».

Конференция состоится в Перми в декабре 1993 года. К участию приглашаются научные работники и преподаватели вузов, работающие над проблемами информатизации и компьютеризации образования, работники Управлений и Департаментов образования, ответственные за решение указанных проблем в своих регионах.

На конференции предполагается выделить следующие направления:

1. Региональные особенности и состояние информатизации образования; организационные формы.

2. Подготовка специалистов:

- * учителей информатики;
- * организаторов работы кабинетов вычислительной техники;
- * учителей-предметников;
- * работников управления.

3. Научно-методическое и учебно-методическое обеспечение процесса компьютеризации образования:

- * разработки учебной и методической литературы;
- * «электронные учебники»;
- * разработки инструментальных педагогических программных средств;
- * разработки средств поддержки локальных сетей школьных КУВТ;
- * проблемы компьютерной (информационной) технологии обучения.

4. Региональные подходы к проблеме информационного обеспечения системы образования:

- * развитие телекоммуникаций;
- * банки педагогической информации, инновационных данных.

5. Разработки региональных компонент предметных стандартов образования «компьютерные» составляющие.

Предполагается:

- * издание сборника тезисов докладов к началу работы конференции;
- * издание сборника трудов по итогам конференции;
- * представление возможностей для демонстрации программных разработок;
- * выставка-продажа учебной и учебно-методической литературы, представленной участниками.

Заявка на участие в работе конференции должна быть направлена до 1 августа 1993 года.

В заявке указывается:

- * Ф.И.О. участника;
- * место работы, должность;
- * ученая степень, звание;
- * область научных или организационных интересов;
- * предполагаемая тема доклада, соавторы;
- * адрес для переписки, телефон, факс, телетайп, E-mail.

ЗАЯВКИ НАПРАВЛЯТЬ ПО АДРЕСУ:
614600, Пермь, К.Маркса, 24, Пединститут.
Оргкомитет конференции РЕГИНФОРМ,
тел. (342-2) 32-75-73, 39-40-20
факс: (342-2) 33-59-23
тайп 134584 Бриз
E-mail: root@pspi.perm.su.
stad@ipkro.perm.su

СЛЕДУЮЩЕЕ СООБЩЕНИЕ - В СЕНТЯБРЕ 1993 ГОДА.

Фирма «СНЕК» сообщает

о создании нового объединения программистов "THE LAND OF DREAMS" - "ЦАРСТВО ГРЕЗ".

В него вошли: компания «SOFT OF FUTURE» - известная по программам EXOLON, FIST, PEEP-SHOW, STRIP-POKER; Борис Бортник (BILCORP); DMW Inc.

Этим объединением подготовлен первый пакет программ "Царство грез-1".

В пакет вошли игры-сериалы: PHANTIS2, BARBARIAN, JOY BLAD.

Все предлагаемые программы - аналоги одноименных программ со "Спектрума-ZX". Эти игры одни из лучших игр для этого компьютера.

Каждая игра состоит из загрузчика, заставки, игрового блока, нескольких этапов, концовки.

(Всего 25 файлов).

Со 2 апреля 1993 года начнется продажа нового комплекта программ Бориса Бортника и его BIL Corp.

Игровой комплект называется
"SPIKY HAROLD".

В него вошел новый "Суперсериал" из 30 файлов. В программах использована оригинальная музыка, великолепная и разная во всех файлах графика, динамические "мультизаставки".

На вашей планете из-за загрязнения произошла мутация. Все роботы, компьютеры, другие обитатели планеты стали вашими врагами.

Вы можете спасти планету, пробравшись в центр погибшей цивилизации и преодолевая все опасности. Чтобы сделать это, нужно пройти вначале по поверхности планеты, затем по подземелью. В конце игры - реклама следующих программ.

С 01.05.1993 г. начинаем продажу хорошо известного вам по многочисленным рекламам игрового пакета "ТЕРМИНАТОР-2".

В пакет вошли примерно 7 программ, объединенных одним сюжетом и написанных объединением "SMASH", известным по программам INDIANA JONES, STREAP SHOW, DREAM, ROBOCORP.

По словам авторов, "сериал "ТЕРМИНАТОР-2" - это более высокий уровень по сравнению с их предыдущими играми".

Все эти комплекты вы можете заказать прямо сегодня!

Примерная цена (рассчитанная на 01.02.1993 г.) комплекта, записанного на дискете или качественной кассете (100% комплектующих фирмы BASF) - 600 рублей, включая почтовые расходы и авторский гонорар. Кассеты рассылаются наложенным платежом.

**ОДНОВРЕМЕННАЯ РАС-
СЫЛКА КАССЕТ ГАРАНТИ-
РУЕТ ПОЯВЛЕНИЕ ИГР
ОДНОВРЕМЕННО И В МОСК-
ВЕ И НА КАМЧАТКЕ.**

Внимание!

Фирма «СНЕК» объявляет о новой системе выплаты авторского гонорара. Желая обеспечить авторам получение достойного гонорара, стимулировать создание новых игр, мы предлагаем вам следующее:

Высылайте на наш адрес (142407, Московская область, г. Ногинск-7, а/я 53) заявку, в которой укажите, какие комплекты вы хотите приобрести. К заявке должны быть приложены квитанции к следующим почтовым переводам:

1. На адрес фирмы «СНЕК»: 150 руб. за каждый комплект (залоговая сумма на почтовые расходы)

2. На адреса авторов (см. ниже) каждого заказываемого комплекта: по 100 руб. за каждый комплект (авторский гонорар).

АДРЕСА для перечисления денег почтовыми переводами авторам:

за комплект "Царство грез-1" - 142407, Московская область, г. Ногинск-7, а/я 77;

за комплект "SPIKY" - 113628, Москва-628, а/я 16, Бортнику Б.

за комплект "ТЕРМИНАТОР-2" - 614010, Пермь, а/я 6703, Маркову М.Ю.

БЕЗ КВИТАНЦИЙ О ПОЧТОВЫХ ПЕРЕВОДАХ ЗАКАЗЫ НА КОМПЛЕКТЫ "SPIKY" И "ТЕРМИНАТОР-2" ПРИНИМАТЬСЯ НЕ БУДУТ!

Примечание для тех, кто читает эту рекламу уже после того, как эти игры «разошлись»: если вы приобрели эти программы не у фирмы «СНЕК» и они вам понравились, перечислите указанную сумму на адрес авторов.

**Мы призываем ценить
труд программистов!**

М. КОВТУН

Программирование в системе Express Pascal

Процедуры и функции

Часто в программах возникает необходимость выполнять в разных местах одни и те же действия — быть может, несколько модифицированные. Например, в программе, выполняющей тригонометрические вычисления, нужно вычислять значения различных тригонометрических функций — синуса, косинуса, тангенса, котангенса и т. д. Или в программе, ведущей диалог с пользователем, нужно выводить на экран сообщения об ошибках (собственно вывод текста сообщения выполняется стандартной процедурой Write, но, кроме вывода текста, его нужно бывает соответствующим образом оформить — заключить в рамочку, добавить какую-либо стандартную часть, дождаться нажатия какой-либо клавиши на клавиатуре и т. п.). Для облегчения решения таких задач предназначены процедуры и функции.

Чем отличаются процедуры от функций? Вы, наверное, заметили различие в двух примерах предыдущего абзаца. То, что вычисляет значение тригонометрической функции, должно выработать результат — вычисленное значение. Такие действия выполняются функциями. При выводе сообщения на экран необходимо выполнить какие-то действия без получения результатов вычисления. Такие действия выполняются процедурами.

В Паскале процедуры и функции оформляются способом, очень похожим на способ оформления программ: они представляют собой как бы маленькие программки, вставленные в большую программу (в некоторых других языках программирования они даже называются подпрограммами).

Описание процедуры или функции помещается в раздел описаний (туда же, где находятся описания констант, переменных, типов, меток). Описание процедуры или функции начинается с заголовка. А то, что идет после заголовка, устроено точно так же, как и программа: сначала идет раздел описаний, а за ним — раздел операторов. Единственное отличие от программы — после зарезервированного слова `end`, заканчивающего раздел операторов, в процедуре или функции должна стоять точка с запятой (в программе там должна стоять точка).

В разделе описаний процедуры или функции может находиться все то, что может быть в разделе описаний программы: описания констант, переменных, типов, меток, и даже описания процедур и функций. Но важное отличие объектов, описанных внутри процедуры или функции, от объектов, описанных в главной программе, состоит в том, что они видны только внутри текущей процедуры или функции, и не видны в других процедурах и функциях и в главной программе. Однако внутри процедуры видны все объекты, описанные вне ее. Объекты внутри процедуры могут иметь имена, совпадающие с именами объектов вне ее; в таком случае внешние объекты не будут видны внутри процедуры (их имена будут «экранированы» внутренними именами). Объекты, описанные внутри процедуры или функции, называются «локальными» (по отношению к данной процедуре или функции), а описанные вне ее — «глобальными» (по отношению к данной процедуре или функции).

Заголовок процедуры имеет вид:

`procedure «имя»—процедуры» («список—параметров»);`

`^ procedure ^` — это зарезервированное слово, отмечающее начало описания процедуры.

33

* Окончание. Начало см.: Информатика и образование. — 1992 (№ 1—5); 1993 (№ 1).

«Имя—процедуры» — это произвольное имя (идентификатор), с помощью которого вы можете в дальнейшем сослаться на эту процедуру. О «списке—параметров» мы поговорим чуть позже; пока что отметим, что он может отсутствовать, и в таком случае окружающие его скобки должны быть опущены.

Заголовок функции имеет вид:

function *«имя—функции»* («список—параметров»): *«тип—результата»*;

^function^ — зарезервированное слово, отмечающее начало описания процедуры. *Имя — функции* — это произвольное имя (идентификатор), с помощью которого можно в дальнейшем сослаться на эту функцию. «Список—параметров», как и у процедуры, может отсутствовать; в этом случае должны быть опущены и окружающие его скобки. «Тип—результата» есть имя типа (например, integer, real и т. п.) — типа значения, возвращаемого функцией.

34 Теперь о списке параметров. Для выполнения своих действий процедуре или функции могут быть нужны аргументы. Например, функция вычисления тангенса должна получить значение угла, для которого вычисляется тангенс; процедура выдачи сообщения об ошибке должна получить текст сообщения, которое оно должна вывести. А некоторые процедуры или функции могут не иметь явных аргументов: например, процедура, выполняющая сброс в начальное состояние (очистить экран, присвоить переменным начальные значения и т. д.), может не иметь аргументов.

Список параметров и описывает набор аргументов процедуры или функции. По сути дела, он является описанием специальных локальных переменных и оформляется в виде, очень похожем на вид описания переменных. Список параметров состоит из последовательности блоков, отделяемых друг от друга точкой с запятой. Каждый из блоков имеет вид:

«имя—переменной», ..., «имя—переменной»: «имя—типа»

или: *var «имя—переменной», ..., «имя—переменной»: «имя—типа».*

Параметры, описанные первым способом, называются «параметрами, передаваемыми по значению»; описанные вторым способом — «параметрами, передаваемыми по ссылке» (о том, что это значит, мы поговорим несколько позже).

Далее в остальной части процедуры или функции имена переменных, заданные в списке параметров, могут использоваться как обычные переменные.

Мы научились описывать процедуры и функции. Но появление операторов в описании процедуры еще не означает, что они будут

выполняться. Для того, чтобы процедура или функция была выполнена, ее нужно «вызвать».

Вызов процедуры выполняется с помощью «оператора вызова процедуры». Оператор вызова процедуры состоит из имени процедуры и идущего вслед за ним в скобках «списка фактических параметров». Список фактических параметров состоит из отдельных фактических параметров, отделенных друг от друга запятой. Количество фактических параметров в списке должно в точности совпадать с количеством параметров, указанным в заголовке процедуры. Если в заголовке процедуры параметр описан как передаваемый по ссылке, то соответствующим фактическим параметром должно быть имя переменной соответствующего типа. Если в заголовке процедуры параметр описан как передаваемый по значению, то соответствующим фактическим параметром может быть любое выражение соответствующего типа (в частности, имя переменной).

Использование параметров в теле процедуры в процессе ее исполнения имеет следующий смысл. Любое использование параметра, передаваемого по ссылке, эквивалентно использованию в этом месте переменной, переданной в качестве фактического параметра. В частности, присваивание такому параметру вызовет изменение значения фактического параметра. Параметры же, передаваемые по значению, нужно рассматривать просто как внутренние переменные процедуры, которым перед началом исполнения процедуры были присвоены значения фактических параметров. Если даже фактический параметр для параметра, передаваемого по значению, был переменной, присваивание нового значения параметру внутри процедуры никак не повлияет на значение переменной, указанной в качестве фактического параметра.

Вызов функции внешне выглядит точно так же, как и вызов процедуры: имя функции и за ним в скобках — фактические параметры. Но вызов функции должен быть не отдельным оператором, а операндом какого-либо выражения (при вычислении такого выражения функция будет вызвана, и возвращенный ею результат будет использован для дальнейшего вычисления выражения). Передача параметров функции выполняется точно так же, как и процедуре.

Для того, чтобы функция могла вернуть результат, внутри ее раздела операторов должен находиться по крайней мере один оператор присваивания вида:

«имя—функции» := «выражение»;

Значение выражения, вычисленного при исполнении последнего такого оператора, и бу-

дет результатом функции. Можно подумать, что в теле функции неявно определена переменная, имя которой совпадает с именем функции, и имя функции можно использовать внутри ее раздела операторов как обычную переменную. Но это не так. Имя функции в указанном смысле можно использовать только в левой части оператора присваивания. Использование его в правой части будет иметь совсем другой смысл — рекурсивный вызов функции (что такое рекурсивный вызов, мы в настоящем цикле статей рассматривать не будем).

Рассмотрим пример:

```
var x, y, z : integer;

procedure p1 ( x : integer );
begin
  x := 1;      { здесь выполняется присваивание локальной переменной }
              { - параметру }
  y := 2;      { здесь выполняется присваивание глобальной переменной }
end;

procedure p2 ( var x : integer );
var y : integer;
begin
  x := 1; { здесь выполняется присваивание параметру, передаваемому }
          { по ссылке - и, значит, произойдет изменение переменной }
          { - фактического параметра }
  y := 2; { здесь выполняется присваивание локальной переменной }
end;

function f1 ( t : integer ) : integer;
var z : integer;
begin
  z := t*t;   { здесь изменяется локальная переменная }
  f1 := z+1;  { здесь формируется результат функции }
end;

begin
  x := 11; y := 12; z := 13;
  { здесь переменные имеют значения: x = 11, y = 12, z = 13 }
  p1(5);
  { здесь переменные имеют значения: x = 11, y = 2, z = 13 }
  y := 12;
  { здесь переменные имеют значения: x = 11, y = 12, z = 13 }
  p1(z);
  { здесь переменные имеют значения: x = 11, y = 2, z = 13 }
  y := 12;
  { здесь переменные имеют значения: x = 11, y = 12, z = 13 }
  p2(x);
  { здесь переменные имеют значения: x = 1, y = 12, z = 13 }
  p2(z);
  { здесь переменные имеют значения: x = 1, y = 12, z = 1 }
  y := f1(x+1);
  { здесь переменные имеют значения: x = 1, y = 5, z = 1 }
end.
```

И в заключение несколько замечаний.

Передача параметров по ссылке уместна тогда, когда вызываемая процедура должна изменить значение передаваемой в качестве параметра переменной. Параметры, которые используются для возврата результатов работы, называются выходными. (Вообще говоря, использование выходных параметров позволяет работать только с процедурами, и не использовать функций. Но использование функций часто делает программу более простой и понятной.) Использование выходных параметров очень полезно тогда, когда функция должна вернуть несколько резуль-

35

Более естественный пример использования процедур и функций приведен в примере большой программы, который представлен в одном из следующих разделов.

татов: например, координаты чего-либо на экране — это два числа.

Другой случай, когда полезно использование параметров, передаваемых по ссылке —

это параметры, занимающие много памяти (например, большие массивы). При передаче по их значению потребуются зарезервировать память для массива не только в вызывающей программе, но и в процедуре: кроме того, будет затрачено большое время на копирование параметра. При использовании передачи по ссылке будет скопирован только адрес массива.

Как уже отмечалось, процедуры и функции полезно использовать тогда, когда одни и те же действия необходимо выполнять в разных местах программы. Кроме этого, в процедуре полезно выделять и неоднократно выполняемые действия — когда они выделены по смыслу. (Например, начальную инициализацию переменных удобно оформлять в виде отдельной процедуры.) Хотя это и удлиняет программу, программа становится понятней. Один из принципов структурного программирования говорит, что каждая процедура (и в том числе главная программа) должна занимать не более одной страницы; если процедура получается длиннее, то стоит ее разбить на несколько процедур. Но при выполнении такого разбиения нужно, разумеется, следить за тем, чтобы каждый из кусков действительно выделялся по смыслу выполняемых им действий.

36

Работа с текстовым и графическим экраном

Практически любая программа должна как-то общаться с пользователем, получать от него указания и выдавать результаты своей работы. И в первую очередь информация о работе программы выводится на экран.

К сожалению, способы вывода информации на экран различны для различных типов компьютеров. Поэтому процедуры Паскаля, обеспечивающие доступ к экрану, несколько отличаются в реализациях для различных типов компьютеров. Здесь мы подробно опи-

```
GotoXY ( 0, 0); { помещает текущую позицию в левый верхний угол экрана }
GotoXY (15, 0); { помещает текущую позицию в левый нижний угол экрана }
GotoXY ( 0, 63); { помещает текущую позицию в правый верхний угол экрана }
GotoXY (15, 63); { помещает текущую позицию в правый нижний угол экрана }
GotoXY ( 7, 31); { помещает текущую позицию в центр экрана }
```

шем процедуры работы с экраном в системе Express Pascal для «Корвета». Знакомство с ними поможет понять принципы построения такой библиотеки процедур, и в будущем вам будет легче разобраться со способами вывода на экран на другом компьютере.

Существуют два (аппаратно реализованных) способа вывода информации на экран: текстовый и графический. Соответственно мы будем говорить о (логических) текстовом и

графическом экранах. Отличительной особенностью «Корвета» является возможность одновременного показа на физическом экране содержимого текстового и графического экранов. (Такое невозможно, например, на IBM PC. Если там вам понадобится вывести одновременно какое-либо графическое изображение и текстовое пояснение к нему, то придется рисовать буквы как графические изображения.)

Текстовый экран «Корвета» состоит из 16-ти строк: каждая из строк состоит из 64-х знакомест — позиций, в которых может находиться литера. Всего имеется 256 картинок литер: в «Корвете» они фиксированы (не могут быть изменены программой). Строки нумеруются числами от 0 до 15; знакоместа в строке — числами от 0 до 63.

Вывод на текстовый экран достаточно прост. С основной процедурой вывода информации — Write и ее модификацией WriteLn — мы уже познакомились. Кроме нее, имеются еще две вспомогательные процедуры. Первая из них — ClrScr — не имеет параметров; ее действие состоит в очистке экрана и установке текущей позиции (позиции, начиная с которой будут размещаться литеры при выводе их на экран с помощью процедуры Write) в левый верхний угол экрана.

Вторая процедура позволяет устанавливать текущую позицию в произвольное место экрана. Формат ее вызова таков:

```
GotoXY ( x, y );
```

Ее параметры (x и y) должны быть выражениями целого типа. Они определяют новое положение текущей позиции на экране. Значение параметра x (горизонтальная координата) должно лежать в интервале от 0 до 63, значение параметра y (вертикальная координата) — в интервале от 0 до 15. Примеры:

В системе Express Pascal текст выводится на экран всегда белым цветом (на черно-белом мониторе — самой яркой градацией серого). Можно изменить цвет текста, но это уже выходит за пределы настоящего цикла статей.

Графический экран состоит из отдельных точек (их еще называют пикселями — английское слово pixel). Эти точки занимают 256 строк, по 512 точек в строке. (Каждая

литера на текстовом экране тоже строится из таких точек, одно знакомство занимает 8 точек по горизонтали и 16 точек по вертикали.) Строки нумеруются числами от 0 до 255 (строка с номером 0 — верхняя строка экрана, строка с номером 255 — нижняя), точки в строке нумеруются от 0 до 511 (точка с номером 0 — левая точка строки, точка с номером 511 — правая точка строки).

Каждая точка может иметь один из восьми цветов; цвета точек нумеруются числами от 0 до 7. Эти числа кодируют следующие цвета: 0 — черный, 1 — синий, 2 — зеленый, 3 — голубой, 4 — красный, 5 — фиолетовый, 6 — желтый, 7 — белый. На черно-белом мониторе цвета передаются полутонами (градациями серого). Яркость точки возрастает с возрастанием номера цвета от черного (цвет 0) до белого (цвет 7).

Имеется семь процедур работы с графическим экраном. Во всех этих процедурах координаты могут быть заданы выражениями целого типа. Если координаты какой-то части фигуры лежат за пределами экрана, соответствующая часть фигуры не будет нарисована.

1. Очистка графического экрана. Формат вызова:

```
ClrGScr;
```

Эта процедура вызывает очистку графического экрана и подготовку его к работе. Процедура должна быть обязательно выполнена до любой другой процедуры вывода на графический экран — иначе ничего на экране не появится.

2. Установка текущего цвета. Формат вызова:

```
SetColor ( clr );
```

clr должно быть выражением целого типа, принимающим значение в интервале от 0 до 7. Изображения, создаваемые остальными процедурами, будут рисоваться цветом, определенным этой процедурой. Если различные изображения должны рисоваться различными цветами, то перед вызовом каждой процедуры вывода изображения должна вызываться процедура установки цвета.

3. Чтение текущего цвета точки. Формат вызова:

```
GetPixel ( x, y )
```

Это функция: ее результат есть цвет (число в диапазоне от 0 до 7) заданной точки. x и y задают координаты точки, цвет которой должен быть прочитан.

4. Рисование точки. Формат вызова:

```
PutPixel ( x, y );
```

Эта процедура закрашивает точку с координатами (x, y) цветом, установленным процедурой SetColor.

5. Рисование прямой линии. Формат вызова:

```
Line ( x1, y1, x2, y2 );
```

Эта процедура проводит прямую линию от точки с координатами (x1, y1), до точки с координатами (x2, y2). Линия проводится цветом, установленным процедурой SetColor.

6. Рисование прямоугольника. Формат вызова:

```
Rectangle ( x1, y1, x2, y2, fill );
```

Эта процедура рисует прямоугольник, два противоположных угла которого (может быть задана любая пара противоположных углов) имеют координаты (x1, y1) и (x2, y2). Пятый параметр — fill должен быть выражением логического типа. Если его значение есть false, то рисуется контур прямоугольника, если true — рисуется закрашенный прямоугольник. Прямоугольник рисуется цветом, установленным процедурой SetColor.

7. Рисование окружности или круга. Формат вызова:

```
Circle ( x, y, r, fill );
```

Эта процедура рисует окружность с центром в точке с координатами (x, y), и радиусом r. Четвертый параметр — fill — должен быть выражением логического типа. Если его значение есть false, то рисуется контур (окружность), если true — рисуется закрашенный круг. Окружность (или круг) рисуется цветом, установленным процедурой SetColor.

В качестве примера приведем программу, рисующую красивый узор на графическом экране. Она использует только часть описанных процедур.

```
var i : integer;
begin
  ClrGScr; ClrGScr;
  for i:=0 to 511
  do begin
    SetColor ( (i mod 2)*7 );
    Line ( i, 0, 511-i, 255 );
  end;
  for i:=0 to 255
  do begin
    SetColor ( (i mod 2)*7 );
    Line ( 0, i, 511, 255-i );
  end;
end.
```

Маленькое замечание. Выражение, являющееся аргументом процедуры SetColor ((i mod 2)* 7) принимает значение 0 для

четных i и значение 7 для нечетных. Таким образом, четные линии рисуются черным цветом, а нечетные — белым. Попробуйте поменять способ вычисления цвета — получатся тоже приятные узоры.

8. Пример большой программы

В этом, заключительном, разделе цикла мы приведем пример относительно большой программы — игры «Минер». После того, как вы разберетесь, как устроена эта программа, можно ввести ее в компьютер, скомпилировать, и в награду на проделанные труды немного поразвлечься.

Правила игры «Минер» такие. Игровое поле состоит из клеток, 11 рядов по 14 клеток в каждом. В некоторых клетках находятся мины (при каждой новой игре мины располагаются в новых местах). Задачей игры является определение клеток, в которых находятся мины. Одна из клеток отмечена

курсором. Вы можете перемещать курсор по игровому полю с помощью клавиш со стрелками. Вы можете открыть клетку, нажав возврат каретки. Если в открываемой клетке оказалась мина, вы погибли, игра оканчивается. Если же в клетке нет мины, вам выдается число — количество мин в восьми соседствующих с открытой клеткой. Эта информация и помогает вам определять, в каких клетках находятся мины. Если вы установили, что в какой-то клетке находится мина, поставьте на эту клетку курсор и нажмите пробел: в клетке появится звездочка. Ваша пометка: «Здесь мина!». Ошибочную пометку можно убрать, повторно нажав пробел. Игра заканчивается успехом, если вы отметили все клетки с минами и открыли все клетки без мин. Для завершения работы с программой нужно нажать Esc.

А теперь — текст программы:

38

```

{-----}
{
{           И г р а           " М И Н Е Р "           }
{
{-----}

const  nr_col = 14;  nr_ln = 11; { Размеры поля по вертикали }
                                { и горизонтали }
      nr_mines = 30;   { Количество мин }

var  x, y : integer; { Координаты текущей клетки }
     c   : char;   { Для ввода литеры с клавиатуры }
     n   : integer; { Число открытых клеток }
     m   : integer; { Число отмеченных мин }

mines : array [ 0..nr_col+1, 0..nr_ln+1 ] of boolean;
        { Наличие мин на поле: }
        { true  - в клетке есть мина, }
        { false - в клетке мины нет. }
        { Этот массив содержит дополнителъ- }
        { ные клетки (отсутствующие в игро- }
        { вом поле - с x-координатой 0 или }
        { nr_col+1 или с y-координатой 0 или }
        { nr_ln+1) - для упрощения процедуры }
        { подсчета числа мин на соседних }
        { клетках. }

state : array [ 1..nr_col, 1..nr_ln ] of byte,
        { Текущий статус клетки: }
        { 0..8 - клетка открыта, число по- }
        { казывает количество мин в }
        { соседних клетках }
        { 9 - клетка отмечена игроком }
        { как содержащая мину }
        { 10 - клетка не открыта }

```



```

{ "*****"
{   Инициализация
}
}

```

```

procedure Init;
var x, y : integer;
    m : integer;
begin
  for x:=0 to nr_col+1      { Везде отметить:      }
  do for y:=0 to nr_ln+1   { "мин нет"      }
  do mines[x,y] := false;
  for x:=1 to nr_col      { Везде отметить:      }
  do for y:=1 to nr_ln   { "клетка не открыта" }
  do state[x,y] := 10;
  Randomize;                { Расставить nr_mines мин в }
  m := 0;                   { случайно выбранные клетки }
  repeat
    x := Random(nr_col)+1; y := Random(nr_ln)+1;
    if not mines[x,y]
    then begin mines[x,y] := true; m:=m+1; end;
  until m = nr_mines;
end;

```

39

```

{ "*****"
{   Нарисовать пустое поле и сообщения-подсказки.
{   Игровое поле имеет nr_col клеток по горизонтали и nr_ln
{   клеток по вертикали. Каждая клетка имеет высоту 16 графич-
{   ческих точек по вертикали (высота одного текстового знако-
{   места) и 24 точки (ширина трех текстовых знакомест).
}
}

```

```

procedure DrawField;
var i : integer;
begin
  SetColor(7);      { сетка рисуется белым цветом }
  for i:=0 to nr_ln { рисуем горизонтальные линии }
  do Line ( 0, i*16, nr_col*24, i*16 );
  for i:=0 to nr_col { рисуем вертикальные линии }
  do Line ( i*24, 0, i*24, nr_ln*16 );
                                { Подсказка:      }
  GotoXY(0,12); Write('Используйте клавиши:');
  GotoXY(2,13); Write
    ('стрелки: перемещение текущей позиции');
  GotoXY(2,14); Write
    ('пробел: отметить мину / убрать отметку');
  GotoXY(2,15); Write
    ('возврат каретки: открыть клетку');
  GotoXY(55,13); Write('Esc -');
  GotoXY(55,14); Write('конец');
  GotoXY(55,15); Write('игры. ');
                                { Заготовка для текущей информации: }
  GotoXY(49,0); Write('Осталось');
  GotoXY(49,1); Write('необследованных');
  GotoXY(49,2); Write('клеток: ');
  GotoXY(49,5); Write('Осталось');
  GotoXY(49,6); Write('ненайденных');
  GotoXY(49,7); Write('мин: ');
end;

```

```
{ "*****"
  {   Функция подсчета числа мин на соседних клетках   }
```

```
function NearMines ( x, y : integer ) : integer;
  var m : integer;
  begin
    m:=0;
    if mines[x-1,y-1] then m:=m+1;
    if mines[x ,y-1] then m:=m+1;
    if mines[x+1,y-1] then m:=m+1;
    if mines[x-1,y ] then m:=m+1;
    if mines[x+1,y ] then m:=m+1;
    if mines[x-1,y+1] then m:=m+1;
    if mines[x ,y+1] then m:=m+1;
    if mines[x+1,y+1] then m:=m+1;
    NearMines := m;
  end;
```

```
label Start;
```

40

```
begin
```

```
Start:
```

```
  ClrGScr; ClrScr;      { Очистить экран, }
  Write(#27';');      { погасить курсор }
```

```
  Init;      { Инициализировать описание минного поля }
```

```
  DrawField; { Нарисовать пустую сетку на экране }
```

```
      { Подготовка главного цикла }
```

```
  x:=1; y:=1; n:=0; m:=0;
  GotoXY(55,3); Write ( nr_col*nr_ln : 4 );
  GotoXY(55,8); Write ( nr_mines : 4 );
```

```
      {----- Главный цикл -----}
```

```
repeat
```

```
  { Вывести указатель (закрасить текущую клет- }
  { ку). дождаться ввода с клавиатуры, стереть }
  { указатель } }
```

```
  SetColor(1);
```

```
  Rectangle( (x-1)*24+1, (y-1)*16+1, x*24-1, y*16-1, true );
```

```
  c := ReadKey;
```

```
  SetColor(0);
```

```
  Rectangle( (x-1)*24+1, (y-1)*16+1, x*24-1, y*16-1, true );
```

```
      { Разбор случаев ввода: } }
```

```
case c of
```

```
  { Стрелки - изменить текущие координаты } }
```

```
  #$1D {влево} : if x=1 then x:=nr_col else x:=x-1;
```

```
  #$1C {вправо} : if x=nr_col then x:=1 else x:=x+1;
```

```
  #$1E {вверх} : if y=1 then y:=nr_ln else y:=y-1;
```

```
  #$1F {вниз} : if y=nr_ln then y:=1 else y:=y+1;
```

```
  { Пробел: } }
```

```
  { если клетка закрыта: поставить звездочку }
```

```
  { если в клетке звездочка: закрыть }
```

```
  { иначе: ничего не делать }
```

```

#$20 {пробел} : if (state[x,y] = 10) and (m < nr_mines)
    then begin
        state[x,y] := 9;
        GotoXY ( (x-1)*3+1, y-1 );
        Write('*');
        m := m+1; n := n+1;
    end
else if state[x,y]=9
    then begin
        state[x,y] := 10;
        GotoXY ( (x-1)*3+1, y-1 );
        Write(' ');
        m := m-1; n := n-1;
    end;

    { Возврат каретки: }
    { если клетка открыта или в ней звездочка: }
    { ничего не делать }
    { если клетка закрыта: }
    { если в клетке мина: взорвался - вы- }
    { дать сообщение, дождаться нажа- }
    { тия клавиши, перейти на Start }
    { если в клетке нет мины: вывести в }
    { клетку число мин вокруг }

#$0D {возврат каретки} :
    if state[x,y] = 10
    then if mines[x,y]
        then begin
            SetColor(4);
            Rectangle ( 43*8, 10*16,
                64*8-1, 12*16-1, true );
            GotoXY(47,10);
            Write( 'Вы взорвались!' );
            GotoXY(43,11);
            Write('Нажмите любую клавишу');
            c := ReadKey;
            goto Start;
        end
    else begin
        state[x,y] := NearMines(x,y);
        GotoXY ( (x-1)*3+1, y-1 );
        Write(state[x,y]);
        n:=n+1;
    end;

end{case};
GotoXY(55,3); Write ( nr_col*nr_ln - n : 4 );
GotoXY(55,8); Write ( nr_mines - m : 4 );
until (n = nr_col*nr_ln) or (c = #$1B{Esc});

    { Теперь нужно разобраться, почему мы вышли }
    { из главного цикла, и либо начать игру сна- }
    { чала, либо закончить работу }

if m = nr_mines
then begin
    { Победа - вывести сообщение, дождаться }
    { нажатия клавиши, перейти на Start }
    SetColor(2); Rectangle ( 43*8, 10*16,
        64*8-1, 12*16-1, true );
    GotoXY(48,10); Write( 'Вы победили!' );
    GotoXY(43,11); Write('Нажмите любую клавишу');
    c := ReadKey;
    goto Start;
end

```

```

else begin
    { Мы попадаем сюда, если нажат Esc. Ничего делать }
    { не нужно - программа заканчивает работу. }
end;
end.

```

Замечания к программе

Как сделать так, чтобы мины каждый раз располагались в новых местах? Ответ: их нужно располагать случайно. Для этого и используется процедура *Randomize* и функция *Random*.

Процедура *Randomize* выполняет инициализацию генератора случайных чисел. Если ее не вызывать, то работа генератора случайных чисел будет начинаться из одного и того же состояния, и вы будете получать одно и то же расположение мин.

Функция *Random* должен быть передан параметр — целое число. Результатом этой функции будет также целое число, случайно выбранное из диапазона от 0 до $n-1$ (n -значение параметров).

Если вам удалось понять, как работает эта программа, значит наши занятия не прошли даром. Тогда попытайтесь усовер-

шенствовать эту программу. Попробуйте сделать следующее:

поменяйте размеры игрового поля; сделайте так, чтобы размеры игрового поля и количество мин вводились игроком в начале работы программы;

если открытая клетка содержит 0, то большого ума не требуется, чтобы понять, что в соседних клетках мин нет, и их можно спокойно открыть. Сделайте так, чтобы после открывания клетки с числом 0 все соседние клетки открывались автоматически;

если, например, открытая клетка содержит 1 и в одной из соседних клеток вы уже заметили мину, то все остальные соседние клетки можно спокойно открывать. Сделайте так, чтобы эти клетки можно было открыть нажатием одной специально отведенной для этого клавиши (например, клавиши табуляции).

Успехов вам!

П. СТАЦЕНКО, Л. ПОДОЛЬСКИЙ, М. СЕМИОНЕНКОВ
(г. Пушино, ИМПБ РАН)

Интегрированная система программирования Quassic-3 для ДВК-3М, ДВК-4 и «Электроники-85»

Вступление

Несмотря на широкое распространение IBM-совместимых компьютеров, перечисленных в заголовке ЭВМ много, и использоваться они будут еще долго, в том числе в школах, техникумах, вузах. Поэтому наличие удобных средств программирования для этих ЭВМ не потеряло своего значения. Рассматриваемая ниже система программирования, на наш взгляд, позволяет работать на ДВК и «Электронике-85» не менее комфортно, чем в системах Turbo-C и Turbo-Pascal на IBM PC.

В состав системы Quassic-3 входит экранный редактор, компилятор, отладчик, средства поддержки выполнения программ. Система работает в среде ОС RT-11 и ее аналогах.

От других интегрированных систем программирования система Quassic-3 отли-

чается тем, что программе при выполнении доступны все компоненты системы программирования, в том числе редактор и компилятор. Благодаря этому система получает качественно новые возможности.

Quassic-3 можно использовать для обучения программированию, создания программ, поддерживающих учебный процесс, создания программ, обслуживающих научное и технологическое оборудование, а также для других целей.

Стиль работы в системе

Пользователь взаимодействует с системой Quassic-3 через экранный редактор. Редактор работает с текстами. Текст — это последовательность строк, находящаяся в памяти ЭВМ и отображающаяся в окне на экране терминала.

При запуске система открывает несколько текстов. Для наших целей пока достаточно упомянуть два: программный и рабочий тексты. Программный текст используется для создания и хранения пользовательской программы, а рабочий текст направляется на стандартный ввод/вывод.

Для отображения системных текстов может использоваться одно или несколько окон на экране (рис. 1).

Нажимая функциональные клавиши, пользователь вводит команды. Имеются команды для:

- выбора текущего текста,
- редактирования текста,
- управления окнами на экране,
- компиляции и выполнения программы,
- отладки

и некоторые другие.

Опишем на простых примерах, как выглядит работа в системе Quasic-3. Первым примером, который мы рассмотрим, будет программа, выводящая столбик первых ста натуральных чисел.

Для того чтобы ввести исходный текст

```

INT N, SUM
PUT "введи числа", /      !выдать приглашение в рабочий текст
EDIT
SUM=0; ERR=0
LOOP
    GET N                  !ввести число из рабочего текста
    IF ERR EXIT; ENDI     !ERR<>0 означает конец текста
    SUM=SUM+N
ENDL
PUT "сумма =", SUM, /
END
    
```

```

INT I
FOR I=1 TO 100
    PUT I, /
ENDF
END
[ программа ]
    
```

```

94
95
96
97
98
99
100
[ рабочий ]
    
```

программы, перейдем в программный текст и наберем программу. Затем нажмем клавишу <выполнить>. По этой команде программа компилируется и запускается. Если при компиляции будет обнаружена ошибка, то на экран будет выдано сообщение, и курсор в программном тексте встанет на место ошибки. То же самое произойдет при обнаружении ошибки выполнения. Можно исправить ошибку и повторить процедуру.

При выполнении нашей программы в рабочий текст будет выведено 100 строк. Обращаем внимание — строки выводятся в текст, а не прямо на экран терминала, как это происходит в других системах. Строки в рабочем тексте можно просмотреть, отредактировать и, если нужно, сохранить в файле. При этом доступны все выведенные строки. Программа и результат ее работы приведены на рис. 1, 2.

Рассмотрим теперь пример, иллюстрирующий ввод данных. Для этого возьмем программу, которая вводит столбец чисел и печатает их сумму:

Рис. 1. Отображение программного и рабочего текстов в окнах на экране

рабочий текст	окно
1	16
2	17
..	18
.....	-----
..18	..
..17	..
..18	..
.....	-----
....	введи числа
100	10
[рабочий]	20
-----

44 Рис. 2. Вывод чисел

рабочий текст	окно
.....	-----
..140	..
..150	..
..сумма = 1200	..
.....	-----
....	введи числа
100	10
[рабочий]	20
-----

Рис. 3. Ввод чисел

Программа начинает свое выполнение выводом в рабочий текст приглашения — «введи числа». После этого вызывается системная процедура EDIT.

Сразу после вызова EDIT и до возврата из нее пользователь может редактировать содержимое рабочего текста. В нашем случае он вводит необходимые числа. При вводе можно пользоваться всеми средствами экранного редактора, в том числе, скажем, прочесть часть данных из файла.

Когда ввод чисел закончен, нажимается клавиша ВВОД. Это приводит к возврату из EDIT в программу. При возврате курсор автоматически устанавливается туда, где он был на момент вызова процедуры EDIT, в нашем случае — на начало столбца чисел. После этого программа вводит числа из рабочего текста оператором GET, подсчитывает их сумму и печатает результат. Результат работы программы представлен на рис. 3.

Приведенные примеры показывают основное отличие редактора системы Quasic-3 от редакторов в других интегрированных системах. В Quasic-3 работа с содержимым

текста может выполняться не только с терминала, но и из программы: PUT — вводит данные в текст, GET — выводит данные из текста, EDIT — позволяет редактировать содержимое текста с терминала.

Из программы над текстами можно выполнять большое количество различных операций. При этом могут использоваться не только системные тексты, но и тексты, открытые программой. Ниже это будет рассмотрено подробнее.

Выполнение отдельных операторов

Помимо выполнения программы в целом, система Quasic-3 позволяет в диалоге выполнять отдельные операторы языка. Вводить их можно, например, в рабочем тексте. Так оператор PUT A, B, C печатает значения переменных A, B, C, а операторы

A=100; B=200; C=300

присваивают этим переменным новые значения, что обычно полезно при отладке программ. Кроме того, это позволяет исполь-

зовать систему как калькулятор, например так:

PUT EXP(4*ATAN(1.))

В системе имеется возможность связывать отдельные операторы или группы операторов с клавишами. При нажатии такой клавиши операторы немедленно выполняются. Эта возможность фактически позволяет расширять набор команд системы и подстраивать ее под нужды пользователя.

Средства отладки

В системе Quasic-3 для отладки можно использовать точки останова и пошаговое выполнение. Точки останова расставляются прямо в исходном тексте программы нажатием соответствующей клавиши.

Достоинством системы Quasic-3 является то, что при отладке в отдельных окнах можно одновременно видеть исходный текст программы и ее вывод, в том числе графический.

Работа с директориями

Система Quasic-3 позволяет организовать файлы пользователя в древовидную структуру директорий, компенсируя тем самым один из недостатков системы RT-11. При этом система позволяет в экранном стиле просматривать содержимое директорий и выбирать файл при помощи стрелок и курсора.

Директория — это файл, содержащий список других файлов или директорий. В системе имеется текст, содержащий текущую директорию, в этот текст можно войти и работать с содержимым директории. Текст директории разбит по вертикали на две части: слева — поле имен, справа — поле комментариев. Текст может, например, выглядеть так:

DIAL.QDI	стандартные модули поддержки диалога (директория)
SUM.Q	программа суммирования чисел
GRAF.Q	программа построения графика функции
CELL.QDI	клеточный автомат (директория)
GRAF2.QDI	графики и кривые (директория)

Поле комментариев может содержать любую информацию, поле имен содержит имена файлов и директорий. Поле имен защищено от обычных команд редактирования. Имеются команды, позволяющие:

переходить из директории в директорию;
создавать / удалять / переименовывать файлы текущей директории;

загружать / сохранять содержимое программного и других системных текстов в файлах.

Механизм диалоговой работы с директориями может использоваться и в программе пользователя. Об этом будет рассказано ниже.

Стиль работы с директориями в системе Quasic-3 был заимствован из редактора Микро-Мир [1].

Язык программирования

Язык программирования системы Quasic-3 — это язык высокого уровня общего назначения.

Язык поддерживает следующие типы данных: символьный, целый, вещественный, указатели, массивы с любым числом измерений. В выражениях могут использоваться арифметические, логические, битовые операции и отношения.

Имеются следующие управляющие конструкции: IF с ELSIF, WHILE, REPEAT, LOOP, FOR, GOTO. Язык поддерживает процедуры с параметрами и модули.

В языке предусмотрены средства для работы с внешними устройствами на физическом уровне, а также аппаратурой в стандарте КАМАК.

Язык является существенным развитием своего предшественника — языка Quasic-2 [2].

Доступ к системе из программы

Система Quasic-3 предоставляет программу богатый сервис, особенно для поддержки диалога. Это в значительной степени является следствием того, что программа имеет доступ к таким компонентам системы программирования, как редактор, средства диалоговой работы с директориями, компилятор.

Далее мы в отдельных пунктах подробно рассмотрим эти вопросы.

Ввод/вывод

Важным элементом идеологии системы Quasic-3 является единообразная работа с

различными объектами ввода/вывода, такими, как окна на экране терминала, тексты, файлы на диске. Работа с этими объектами поддерживается специальными процедурами, экспортируемыми системой, называемыми драйверами ввода/вывода. Имеются следующие драйверы: окон, текстов, файловой системы, клавиатуры, каждый из которых поддерживает работу с соответствующими логическими объектами.

Доступ к каждому объекту осуществляется через канал — целочисленную переменную в программе пользователя. При открытии объекта в канал заносится идентификатор объекта.

Приведем примеры операций ввода/вывода:

```

OPEN #CH,#FDRV,"A.DAT"  !открытие файла (#CH - адрес канала,
                        !#FDRV - адрес драйвера файловой сис-
                        !темы, "A.DAT" - имя файла)
46 READLN CH,#BUF,80    !чтение строки по каналу CH
WRITLN CH,#BUF          !запись строки по каналу CH
CLOSE #CH               !закрытие канала CH

```

Драйверы совместимы на уровне операций последовательного ввода/вывода. Операторы форматного ввода/вывода в языке реализованы на базе операций последовательного ввода/вывода, поэтому они могут в равной степени использоваться для ввода/вывода в файл, окно, текст.

Помимо общих операций, драйверы могут иметь и свои специфические. Например, у драйвера текстов (редактора) их более 60.

Драйвер окон. Работа с окнами поддерживается драйвером окон, обеспечивающим вывод в окно графической и текстовой информации. Драйвер поддерживает работу как с цветным, так и черно-белым терминалом.

Окно — это прямоугольная часть экрана. Размеры окна и его положение на экране задаются при его открытии и в дальнейшем изменены быть не могут. Окна могут пересекаться произвольным образом, однако этот факт при выводе никак не учитывается: изображение в области пересечения является результатом вывода во все пересекающиеся окна. Фактически это означает, что окна не должны пересекаться. Это ограничение, однако, в большинстве случаев вполне приемлемо.

```

VIS   CH,CH2           !делает текст CH видимым в окне CH2
                        !(сразу после создания текст неви-
                        !дим)

```

Если при выводе в окно происходит выход за его пределы, то вышедшая часть изображения отсекается.

В окне могут использоваться три системы координат: пиксельная, символьная, математическая. В символьной системе координат окно рассматривается как совокупность знакомест, в пиксельной системе — как совокупность пикселей (точек), в математической системе координат окно соответствует прямоугольнику математической плоскости. Пользователь определяет математическую систему координат, исходя из содержания решаемой задачи.

Драйвер поддерживает вывод точек, отрезков, прямоугольников и окружностей, а также вывод текстовой информации. Мож-

но закрашивать области произвольной формы и работать с прямоугольными фрагментами изображения. Поддерживаются спрайты и курсоры, определяемые пользователем.

Драйвер окон базируется на графическом пакете ГРАН [3].

Драйвер текстов. Одной из самых важных черт системы Quasic-3 является то, что в ней редактор оформлен как драйвер текстов. В системе Quasic-3 экранный редактор играет приблизительно ту же роль, что и драйвер терминала в других системах. Благодаря этому взаимодействие пользователя с системой и программой при ее выполнении проходит в стиле, похожем на работу в экранном редакторе.

Драйвер текстов поддерживает работу с текстами. Текст — это последовательность строк, находящаяся в памяти ЭВМ. Количество строк в тексте переменное и ограничивается только количеством доступной памяти.

Для работы с текстами имеется большой набор операций, включающий как стандартные операции последовательного ввода/вывода, так и специфические операции драйвера текста. Приведем некоторые примеры:

NVIS	CH	!делает текст CH невидимым (с невидимым текстом можно работать из программы, но его содержимое на экране не отображается)
POSIT	CH,P,L	!устанавливает курсор в тексте CH
WRITLN	CH,"Y-"	!выводит строку в текст
READLN	CH,#BUF,BUFSZ,#CNT	!читает строку из текста
FIND	CH,"SIN(X)"	!ищет образец в тексте
IL	CH	!вставляет пустую строку в текст
COPFG	CH,LB,LE,CH2	!копирует в текст CH из текста CH2 строки с номерами от LB до LE
LOAD	CH,"ABC.TXT"	!загружает в текст содержимое файла

Благодаря возможности выполнять операции над текстом из программы драйвер текстов становится мощным инструментом для разработки программ, обрабатывающих текстовые данные.

Среди прочих операций драйвера текста особое по важности место занимает операция FEDIT, предоставляющая пользователю возможность менять содержимое текста с терминала. (Упомянутая ранее процедура EDIT является надстройкой над ней.) Операция имеет такой вид:

FEDIT CH, #CMD.

Сразу после этого вызова и до возврата в программу пользователь может редактировать содержимое текста CH с терминала. Выход из FEDIT происходит при вводе с клавиатуры кода, не являющегося командой редактирования. Этот код помещается в переменную CMD и может быть использован программой.

В ряде случаев бывает необходимо ограничить возможности редактирования тем или иным образом, например, запретить модификацию текста или разрешить редактирование только определенных полей. Для достижения этих целей служат операции MASK и BOUND: операция MASK позволяет запретить любые операции драйвера текстов и соответствующие им команды редактирования, а BOUND — ограничить перемещение курсора по тексту только некоторым прямоугольником.

Драйверы файловой системы и клавиатуры. Драйвер файловой системы поддерживает доступ к файлам и устройствам через систему ввода/вывода RT-11.

Драйвер клавиатуры поддерживает ввод с клавиатуры без отображения вводимых символов на экране. Драйвер клавиатуры используется достаточно редко, так как

основным средством ввода/вывода с терминала является драйвер текстов.

47

Диалоговая работа с файловой системой из программы

При выполнении программы часто возникает потребность ввести в диалоге имя файла для доступа к данным на диске. Непосредственный ввод имени не очень удобен, гораздо удобнее иметь перед глазами список файлов и выбрать курсором нужный. Кроме того, часто возникает потребность, не выходя из программы, в диалоге сменить директорию, переименовать или удалить файл. То есть необходима возможность в диалоге с программой выполнять те же действия, которые можно выполнять с файлами при работе в системе программирования.

Система Quasic-3 позволяет достаточно просто реализовать диалоговую работу с файловой системой из программы. Для этого имеются две подпрограммы: OPENDIR и GETNAME.

Подпрограмма OPENDIR открывает текст для работы с директориями, аналогичный системному тексту, описанному в пункте «Работа с директориями». Текст разбит по вертикали на поле имен и поле комментариев. При выполнении операции FEDIT над таким текстом, помимо команд редактирования, можно выполнять и команды работы с директориями так, как это было описано ранее.

Подпрограмма GETNAME позволяет программе прочитать из текста, содержащего директорию, содержимое поля имени в текущей строке. Зная имя файла, программа может использовать его для чтения или записи своих данных.

Вызов компилятора из программы

В системе Quasic-3 компилятор может вызываться как процедура из программы, эта возможность называется динамической компиляцией. Динамическая компиляция полезна в программах, которые в диалоге вводят формулу или алгоритм, а затем по ним выполняют вычисления.

Вызывается компилятор с помощью процедуры RTCOMP следующим образом:

```
RTCOMP CH, #ADR, #P, #L,
```

где:

CH — канал текста, содержащего компилируемый исходный модуль;

ADR — адрес точки входа в полученный при компиляции код;

P, L — позиция ошибки в тексте (код ошибки в переменной ERR).

48

Перед вызовом RTCOMP программа должна открыть текст и занести в него исходный модуль. Это может быть сделано разными способами: текст может быть прочитан из файла на диске, введен с терминала или сгенерирован программно.

В результате выполнения RTCOMP в памяти создается готовый к выполнению

```
REAL X
```

```
INT L,P
```

```
PROC ADR^ !указатель на код, полученный при компиляции
```

```
LABEL ErrExit
```

```
ERASE TI !очищает рабочий текст
```

```
PUT "X=" !выдает приглашение
```

```
EDIT !пользователь вводит выражение правее X=
```

```
NXLB TI !устанавливает курсор в рабочем тексте в начале следующей строки
```

```
RTCOMP TI,#ADR,#P,#L !компилирует содержимое рабочего текста
```

```
IF ERR GOTO ErrExit; ENDI
```

```
RTCALL TI,ADR,#P,#L !выполняет полученный код
```

```
IF ERR GOTO ErrExit; ENDI
```

```
PUT X !печатает X
```

```
DELRTCOMP !удаляет результат компиляции
```

```
STOP
```

```
ErrExit:
```

```
POSIT TI,P,L !устанавливает курсор на место ошибки
```

```
TYPERR !печатает сообщение об ошибке
```

```
DELRTCOMP !удаляет результат компиляции
```

```
END
```

код. Место для кода система выделяет сама. Адрес запуска кода выдается в выходном параметре ADR.

Динамически компилируемая программа может содержать обращение к любым переменным и процедурам, экспортируемым системой или основной программой.

Полученный RTCOMP код выполняется с помощью процедуры RTCALL, имеющей такой вызов:

```
RTCALL CH, ADR, #P, #L,
```

где:

CH — канал исходного текста, при компиляции которого получен код;

ADR — адрес точки входа в код;

P, L — позиция ошибки выполнения в исходном тексте (код ошибки в переменной ERR).

Подчеркнем, что, если выполняемый код завершится аварийно, RTCALL завершится нормально, передав основной программе информацию об ошибке.

Покажем на простейшем примере использование компилятора в программе. Программа вводит из рабочего текста выражение, вычисляет его и выдает результат снова в рабочий текст.

Примеры.

Выше мы говорили про такие нетривиальные возможности системы Quasic-3, как возможность вызывать из программы компилятор и экранный редактор; сейчас мы рассмотрим примеры программ, где это используется.

Программа построения графиков функций. В качестве первого примера рассмотрим программу построения графиков функций. Программа в диалоге вводит формулу, задающую функцию, и по ней строит график.

При работе программы на экране открыты три окна: для задания формулы, задания границ изменения переменных X , Y и для вывода графика (рис. 4).

Имеется команда, позволяющая переключаться между окном формулы и окном границ. В этих окнах можно, используя команды редактирования, ввести формулу и границы изменения переменных. После того как формула и границы установлены, можно ввести команду, по которой строится график.

Для ввода формулы и границ программа использует драйвер текстов. Программа открывает два текста и делает их видимыми в соответствующих окнах. С помощью операций MASK и BOUND устанавливаются ограничения на команды редактирования так, что можно редактировать только поля правее знаков «=» и нельзя было вставлять и удалять строки. Для редактирования содержимого текстов программа выполняет над соответствующим текстом операцию FEDIT.

Для построения графика из текста границ считываются значения x_{\min} , x_{\max} ,

y_{\min} , y_{\max} . Затем программа компилирует содержимое текста, содержащего формулу. График строится обращением в цикле к коду, полученному при компиляции.

Программа имеет размер всего 180 строк.

На рис. 5 представлена более развитая программа. Она может строить графики функций одной и двух переменных, график в полярных координатах, а также параметрически заданные кривые.

Клеточный автомат. В качестве второго примера приведем программу, где в диалоге надо вводить не формулу, а алгоритм. Это программа «Клеточный автомат».

Программа работает с прямоугольным полем M на N клеток. Верх поля сшит с низом, а левый край — с правым. Состояние каждой клетки задается двумя целочисленными переменными A и B . Поле отображается в окне на экране терминала; если значение A в клетке больше нуля, то клетка закрашивается. Значение B на отображение клетки не влияет.

Задается правило перехода клетки из текущего состояния в следующее. Предполагается, что следующее состояние клетки зависит только от ее текущего состояния и состояния ее ближайших восьми соседей.

После того как задано начальное состояние поля и правила перехода, запускается процесс последовательной смены состояний поля. Задавая разные правила перехода, можно моделировать весьма разнообразные явления. Наиболее известным примером клеточных автоматов является игра «Жизнь».

Правило перехода задается на языке системы Quasic-3. Для доступа к состоянию клетки и состоянию ее ближайших соседей

49

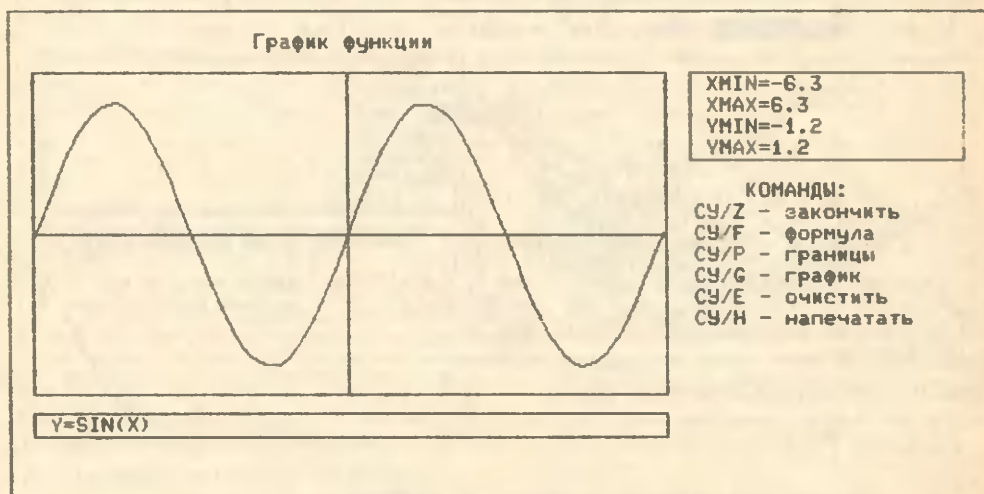


Рис. 4. Программа построения графиков

конец **☐** директория рисовать очистить оси



```

xmin=-2
xmax=2
ymin=-2
ymax=2
a=4
b=0

```

Команды общего назначения

СУ/З, СУ/Н - выбор пункта меню
ВВОД - выполн. пункта меню
СУ/У, СУ/Л - выбор окна !

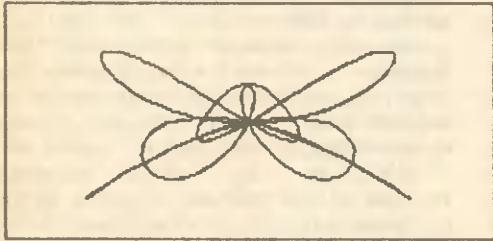
```

y=SIN(a*x)/EXP(x*x)

```

выдать копию экрана на принтер

конец **☐** директория рисовать очистить оси



```

xmin=-1.5
xmax=1.5
ymin=-1.5
ymax=1.5
a=7
b=0

```

Команды общего назначения

СУ/З, СУ/Н - выбор пункта меню
ВВОД - выполн. пункта меню

```

x=SIN(a*t+b)*COS(c*t) !бабочка
y=SIN(a*t+b)*SIN(d*t)

```

выдать копию экрана на принтер

конец **☐** директория рисовать очистить оси



```

xmin=-2
xmax=2
ymin=-2
ymax=2
zmin=0
zmax=10
a=0

```

Команды общего назначения

СУ/З, СУ/Н - выбор пункта меню
ВВОД - выполн. пункта меню

```

z=x*x+y*y

```

выдать копию экрана на принтер

Рис. 5. Программа "Графики и кривые"

используются переменные, представленные следующей таблицей:

ULA, ULB	UA, UB	URA, URB
LA, LB	A, B	RA, RB
DLA, DLB	DA, DB	DRA, DRB

При запуске программы на экране создается шесть окон:

- окно поля;
- окно текущего состояния клетки;
- окно нового состояния клетки;
- окно правил перехода;
- окно директории;
- окно справочной информации

и меню в верхней части экрана (рис. 6).

Окно поля отображает состояние поля. В этом окне можно перемещаться по клеткам. Состояние текущей клетки отображается в окне текущего состояния клетки. Команда <клетка> позволяет установить новое состояние клетки, значения A и B берутся из окна нового состояния клетки. Команда <все поле> устанавливает одинаковое состояние во всех клетках поля. Окно поля с текстом не связано.

В окне нового состояния клетки отображается текст из двух строк, его можно редактировать правее знаков «=», аналогично тому, как это сделано в программе построения графика функции.

В окне правил перехода отображается текст, содержащий алгоритм перехода клетки из текущего состояния в следующее. Текст может содержать любое количество строк и редактироваться без ограничений. Правило перехода компилируется процедурой RTCOMP, а полученный код используется для вычисления нового состояния поля. Смена состояний происходит по команде <ход>.

Находясь в окне директории, можно сохранять состояние программы в файле, загружать новое состояние из файла, переходить из директории в директорию. Работа с директорией реализуется с помощью подпрограмм OPENDIR и GETNAME.

В окне справочной информации виден текст, содержащий описание программы, текст защищен от изменений и может только просматриваться.

Программа «Клеточный автомат» имеет размер 750 строк.

Из приведенных примеров видно, как программа может использовать средства системы программирования, применяемые при разработке программ, редактор, компилятор, средства работы с директориями, help. Благодаря доступу к этим средствам написать программу построения графиков функций или подобную «Клеточному автомату» достаточно просто. В системах, где такого доступа нет, это становится весьма сложной задачей.

Отметим, что в рассматриваемых примерах содержится схема решения не только поставленных задач, но целого класса аналогичных задач. Речь идет о всех случаях, когда программе нужно ввести формулу

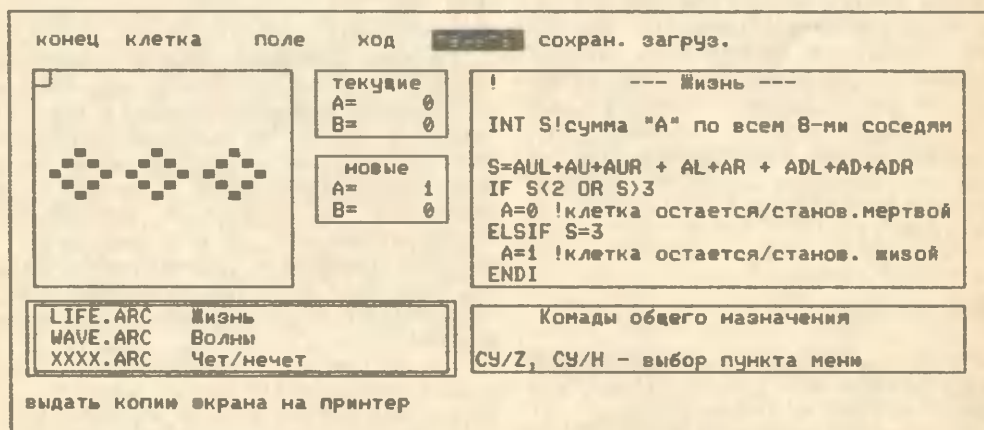


Рис. 6. Программа "Клеточный автомат"

или алгоритм, а затем использовать их для счета. Естественно, формула или алгоритм могут вводиться не только в диалоге, но и, например, читаться из файла.

Систему Quasic-3 можно приобрести по адресу: 142292, Пушкино, Московская область, ИМПБ РАН. Телефоны: Пушкино: 3-25-16; Москва: (095) 923-35-58 (11:15—12:00, 16:00—17:00)

драйвер текста НДТ-83 и системы на его базе // Микропроцессорные средства и системы.— 1987.— № 1.— С. 39—43.

2. Подольский Л. И., Лясковский А. П. Предварительное сообщение о системе Quasic-2 для программирования микро-ЭВМ.— Пушкино: ОНТИ НЦБИ АН СССР, 1990.

3. Семионенков М. Н. Комплекс подпрограмм растровой графики ГРАН для ДВК и «Электроника-60». Версия 2.5.— Пушкино: ОНТИ НЦБИ АН СССР, 1990.

Литература

1. Варсанюфьев Д. В., Дымченко А. Г., Кушниренко А. Г., Лебедев Г. В. Непосредственный

52

А. ЩЕГОЛЕВ
(Москва)

Школьная информатика и язык Пролог

На фоне всеобщего повального увлечения алгоритмизацией, элементами процедурного программирования, системами счисления и т. п. в свое время откровенным диссонансом прозвучали сообщения об использовании в обучении информатике языка Пролог. Стремление ввести еще один язык программирования многим казалось и все еще кажется обременительной роскошью, претенциозным желанием «нового ради нового».

Предваряя публикацию методических материалов по использованию Пролога в обучении, постараемся ответить на вопрос, почему *непроцедурный* язык Пролог (или подобный ему) мыслится одним из необходимых звеньев цепи, именуемой школьной информатикой.

Немного истории

«Основы информатики и вычислительной техники» появились в школах страны, как известно, в 1985 г. Примечательно, что, оглядываясь назад, разные люди (учителя, методисты, ученые, сами школьники и их родители) по-разному оценивают значение то-

го решительного шага, каким стало введение этого нового предмета. Даже среди сторонников школьной информатики нет единства в определении целей и содержания этой учебной дисциплины.

И все же некоторые итоги семилетнего пути сегодня подвести можно. Мы полагаем, что как сторонники, так и противники предмета согласятся с тем, что:

учебные заведения страны постепенно стали оснащаться средствами вычислительной техники. Можно по-разному к этому относиться, но создание подобной дорогостоящей материально-технической базы заставило многих причастных к школе людей задуматься не только о перспективах «компьютеризации», но и о роли каждого учебного предмета в формировании личности школьника, а некоторых привело к размышлениям о судьбах системы образования в целом;

многие учителя, преподававшие «до прихода в информатику» другие предметы, поистине нашли себя в новом качестве. Интерес к работе, развитие собственной личности, возросшее самоуважение этих учителей во многих случаях привели к тому, что и

школы, в которых они работают, существенно изменились. Как бы очнувшись от долгого сна, школа преобразилась, обрела «собственное лицо». Произошло это, естественно, там, где учителя информатики не ограничили сферу своей деятельности компьютерными играми или программистскими упражнениями, а активно искали пути выхода за рамки узкого понимания своего предмета.

В известной мере эти учителя взвалили на свои плечи *общую проблематику* обучения и воспитания: ведь именно они налаживали контакты с преподавателями русского языка, физики, химии и т. д., интуитивно чувствуя, что их сфера деятельности лежит на стыке разных областей знания. Именно они встретились с мощнейшим синдромом «машинобоязни» своих учеников, искали и находили пути преодоления этого «комплекса». Именно они наиболее остро столкнулись с воспитательными проблемами организации увлекающей учеников деятельности (с «перекосами» в поведении «хакеров», например). Именно им приходилось подчас брать на себя руководство совместной работой не только учеников, но и своих коллег (при выпуске школьных газет, альманахов и т. д.).

Именно эти ищущие, мыслящие учителя одними из первых осознали, что рассказы об устройстве компьютера и умение писать примитивные программы на Бейсике, Фокале или Паскале — не лучшее подспорье для вступающих в жизнь. Учитель, не задающий себе извечного вопроса «*Зачем моим ученикам то, чему я собираюсь их научить?*», не может считаться человеком ответственным. А что может быть страшнее безответственного учителя?

К сожалению, осознание ограниченности традиционных представлений о школьной информатике привело многих из лучших (!) учителей к отрицанию необходимости существования этой дисциплины как самостоятельного учебного предмета. Ведь именно они, анализируя свой опыт, пришли к выводу, что *компьютер — не объект изучения, а всего лишь средство, инструмент для решения задач, возникающих в различных сферах человеческой деятельности.*

Присоединяясь к этой (уже достаточно распространенной) точке зрения на использование ЭВМ в учебной деятельности, автор предлагает поразмышлять о том, какие выводы отсюда следуют.

Информатика — «служанка наук»?!

Сформулируем два вопроса, ответы на которые позволяют «предсказать судьбу» информатики как самостоятельного предмета.

Во-первых, разумно ли взваливать на учителей-предметников работу по овладению компьютерным инструментарием для решения задач в *своей предметной области*? Очевидно, это потребовало бы:

а) определения нового содержания и методических принципов преподавания этих предметов, поскольку любое применение компьютера не в качестве калькулятора неизбежно потребует переосмысления дидактических подходов к каждому из данных предметов. Готовы ли к этому авторы программ и учебников по этим предметам?

б) массовой переподготовки учителей-предметников, включающей как переосмысление учителями концепции и методики преподавания предмета, так и приобретения умений уверенного владения средствами вычислительной техники. Готова ли к этому система переподготовки (да и подготовки) педагогических кадров?

в) наличия в каждой школе работника, готового взять на себя нелегкий груз обеспечения технического обслуживания, увы, не слишком надежной техники (и это при том, что сейчас, не жалея подчас личного времени и сил, этим уже занимаются учителя, большинство из которых будет вынуждено уйти из школы, если информатика исчезнет как предмет!). Готова ли к этому школа?

Очевидно, что таким способом проблему не решить. Изменять содержание отдельных школьных предметов, конечно же, придется. Однако вряд ли целесообразно разрушать при этом рожденное в таких муках новое качество — предмет «Основы информатики», а главное, общность учителей, в той или иной степени готовых участвовать в решении проблем информатизации общества.

Во-вторых, если информатике все-таки суждено остаться на правах отдельной учебной дисциплины, то чем ей надлежит заниматься? Ответ как будто очевиден: конечно же, использованием компьютеров для решения задач, возникающих в самых разных областях знания (в частности, в различных школьных предметах). Так ли это? Нет ли у информатики других, более масштабных задач?

Роль «служанки наук» вряд ли может считаться зазорной для этого предмета — ведь мы и не отрицали, что в большинстве случаев знания по информатике имеют вспомогательное (инструментальное) значение. Однако *какие именно задачи*, требующие «вмешательства» информатики, могут решаться в конкретных предметах? Верно ли, что на уроках физики должны решаться специфические «физические» задачи, на уроках химии — ни на что не похожие «химические», на уроках истории (ведь и там решаются задачи!) — «исторические» и т. д.?

Вероятно, у любого мыслящего человека, обдумывающего конкретику такого «вспомогательного» предмета, возникнет потребность выяснить, что общего есть у всех этих задач, а в чем коренятся принципиальные различия в подходах к их решению.

Отсюда следует вывод, что для школьной информатики есть занятие поважнее, нежели применение компьютеров. Ведь чтобы научить решать самые разные задачи, необходимо заняться типологией задач, возникающих в различных сферах человеческой деятельности. Учебная информатика (в этом понимании) все более нацеливается на общие, межпредметные, методологические аспекты, становится своего рода практической философией. При этом очевидно, что «на докомпьютерном уровне» такой предмет существовать бы не смог — именно использование компьютеров оправдывает в глазах учеников деятельность по моделированию самых разных объектов, делает ее наглядной и увлекательной.

Именно информатика должна научить школьников общим принципам решения самых разных задач, возникающих в реальной жизни человека. Естественно, эти принципы должны применяться при решении задач в отдельных учебных предметах.

Отсюда следует парадоксальный на первый взгляд вывод: «информатизация» школьных предметов вовсе не обязательно связана с «компьютеризацией» этих учебных дисциплин. Ведь компьютер — это необходимое средство моделирования явлений различной природы на занятиях по информатике; другие же учебные предметы «пользуются» методами решения задач, освоенными учениками в результате такого моделирования. Поэтому введение компьютеров в тот или иной учебный предмет (кроме информатики) — не самоцель. Более того, можно даже ожидать, что эффективность использования ЭВМ в этих предметах будет существенно ниже, чем в информатике!

Как же использовать компьютеры в информатике?

Традиционный ответ на этот вопрос известен: конечно же, через алгоритмизацию и процедурное программирование. Иногда цель изучения информатики в школе напрямую отождествляют с формированием «алгоритмического мышления», то есть со способностью строить, исполнять, а также анализировать выполнение алгоритмов.

Экстремистские крайности такого подхода иногда демонстрируют использованием выражения «компьютерная культура». Подразумевается, что компьютер порождает не-

кий общекультурный феномен, «просто» человеку недоступный — отсюда и необходимость в специальных занятиях, чтобы человек смог до этой «культуры» дорасти. Для любого же непредвзятого человека очевидно, что нужно не снижать культурный уровень человека до уровня, диктуемого компьютером, а, напротив, последовательно поднимать развитие технологий, связанных с использованием компьютеров, до «человеческого» уровня. *Не подстраивать мышление человека к принципам действия и возможностям компьютеров, а приближать работу компьютера к тому, как мыслит человек!* Алгоритмизация и программирование как раз и призваны дать ученикам возможность моделировать на компьютере самые разные процессы.

Легко ли преподавать алгоритмизацию?

Увы, не секрет, что в обычной (не специализированной) школе алгоритмизация и программирование далеко не всем ученикам даются легко. Временами складывается впечатление, что основы алгоритмизации и элементы процедурного программирования воздвигают на пути учеников достаточно высокий «барьер», преодолеть который не так-то легко.

По нашим наблюдениям, в процессе изучения этих тем в каждом классе отчетливо выявляются три группы учеников.

К первой группе отнесем тех трехчетырех человек, для которых трудностей в программировании, кажется, не существует вовсе. Этим школьников не нужно уговаривать поработать в компьютерном классе во внеурочное время — напротив, надо приложить серьезные усилия, чтобы выставить их из кабинета ИВТ, где они обычно засиживаются допоздна.

Вторая группа учеников не может похвастаться «туннельным эффектом», позволяющим миновать указанный «барьер» без труда. Однако именно эта группа учеников отражает подлинное мастерство учителя, ибо только благодаря его самоотверженным усилиям все эти школьники преодолевают «барьер» алгоритмизации и программирования.

Но есть и *третья группа*. Это те, кто, несмотря на все старания учителя, на педагогические находки и дополнительные занятия, указанный «барьер» так и не преодолевают. Можно аттестовывать этих учеников с положительными оценками, можно закрывать глаза на их реальные «достижения», однако опыт преподавания наводит подчас на безрадостные мысли. Может быть,

алгоритмизация — действительно «не для всех»?

Каков же удельный вес второй и третьей «групп» в конкретном классе? Как уже отмечалось, это напрямую зависит от мастерства и долготерпения учителя. Однако почти в каждом классе доля третьей «группы» составляет 20—60 % всех учеников.

Возникает естественный вопрос: так ли уж необходимо изучение алгоритмизации и программирования, если оно сопряжено со столь серьезными трудностями? Действительно ли «алгоритмическое мышление» необходимо сегодня каждому (ведь речь идет об общеобразовательной школе)?

Роль алгоритмизации в школьной информатике

Для чего же ныне тратится столько сил и времени при изучении основ алгоритмизации и элементов программирования?

1. Овладение основами алгоритмизации учит школьников *планировать деятельность* (свою и чужую). Иными словами, умение принимать решение о том, какие действия и в какой последовательности нужно проделать для достижения цели в зависимости от изменения внешних условий — важнейшее качество, необходимое каждому, какой бы профессиональной деятельностью он ни занимался. Все мы принуждены время от времени принимать подобные решения; научить делать это эффективно — одна из задач общего образования.

2. Владение основами алгоритмизации и программирования позволяет *решать задачи с помощью ЭВМ*. При этом круг предметных областей, с которыми связаны задачи, может быть чрезвычайно широк. Однако необходимость «говорить на языке», навязанном компьютером, вовсе не означает, что этот язык (имеются в виду алгоритмические языки вообще) наиболее удачный. Более того, трудности, с которыми устойчиво сталкивается изучение алгоритмизации, как раз и демонстрируют, что алгоритмический стиль чужд человеческому мышлению. Вопреки распространенной точке зрения, человек почти никогда не мыслит алгоритмически! Нетрудно понять, почему большая часть времени, отводимого на изучение информатики, тратится на овладение «инструментарием» («натаскивание» на алгоритмизацию и программирование).

Есть ли альтернатива?

Можно ли предложить иное средство, позволяющее использовать ЭВМ для реше-

ния вполне «человеческих» задач? Хотелось бы, чтобы это средство не вызывало такого же устойчивого «отторжения» со стороны психики учеников, как это происходит с алгоритмизацией и процедурным программированием.

Очевидно, такую альтернативу следует искать в области иных, *неалгоритмических* принципов «общения» с компьютером. Разницу между алгоритмическим и неалгоритмическим подходами можно понять на следующем простом примере.

Для того чтобы объяснить человеку, как добраться до нужного места в незнакомом городе, можно предложить два способа.

Первый из них заключается в том, чтобы последовательно описать действия этого человека для приближения к интересующему его объекту. Очевидно, что речь идет о составлении (а затем и об исполнении) алгоритма. К сожалению (или к счастью?!), человек — неважный составитель, а тем более исполнитель алгоритмов. Именно этим можно объяснить тот факт, что без дополнительных расспросов и «кружения» по незнакомой местности человек редко находит нужный объект.

Другой способ также общеизвестен. Вместо описания последовательности действий можно нарисовать схему расположения улиц, домов и т. п. в интересующем нас районе. При этом на схеме можно вообще не указывать маршрут: человек в состоянии сделать это и сам (причем, как правило, даже не единственным способом). Таким образом, здесь подразумевается алгоритм (или даже целое семейство алгоритмов), позволяющий добраться до нужного места; однако человеку вручается все-таки описание не самой последовательности действий, а устройства объекта (микрорайона).

Итак, вместо подхода «*Как сделать?*» можно использовать принцип «*Как устроено?*». Первый из этих подходов, очевидно, соответствует алгоритмизации; второй же может быть назван декларативным, неалгоритмическим.

Неожиданно емкую аналогию можно найти даже в столь далекой, казалось бы, от информатики области, как постановка спектаклей. Театральный режиссер, реализуя свой замысел, в принципе может превратить артистов в «исполнителей» детального алгоритма, в котором раскрывается рисунок каждой роли. Однако, как известно, излишне «жесткие» системы обычно оказываются нежизнеспособными. Любая маломальски нештатная ситуация грозит спектаклю провалом.

Однако режиссер может поступить и по-другому. Вместо объяснения конкретных действий он может рассказать артисту о за-

ботах, тревогах, страстях его персонажа. Режиссер вправе рассчитывать, что нужные действия артист найдет сам, исходя из того проникновения во внутренний мир героя, которое произошло в результате подобного рассказа.

Итак, противопоставление «алгоритмического» и «декларативного» подходов налицо. Использование языка Пролог является реализацией декларативного (неалгоритмического) принципа «общения» с ЭВМ.

Собственно о Прологе

Тот, кто, «намучившись» с алгоритмизацией, начинает использовать в учебной деятельности язык Пролог, на определенное время может приобрести эйфорический настрой. Принципы декларативного описания так называемых баз знаний представляются простыми и естественными (что подчеркивает их близость механизмам мышления человека). Кажется, что алгоритмические проблемы, вопросы организации последовательностей действий просто исчезли!

Но в этой легкости заключено коварство «декларативного» подхода. Оказывается, если «забыть» о процедуре, с помощью которой компьютер находит ответы на вопросы, обращенные к базе знаний, простота восприятия Пролога через некоторое время оборачивается «неимоверными сложностями»!

Происходит это по причинам достаточно общего характера. Дело в том, что при традиционном подходе к описанию явлений в виде алгоритма четко описывается (формализуется) процесс деятельности. При этом ожидаемый результат этой деятельности в лучшем случае описывается, что называется, «на пальцах».

При построении баз знаний на Прологе, наоборот, четко описывается именно результат деятельности (в виде базы знаний и вопроса к ней). Что же касается самого процесса достижения этого результата, язык Пролог «не настаивает» на том, чтобы он был достаточно строго формализован.

С этой точки зрения само по себе использование Пролога еще не гарантирует легкости и доступности для учеников моделирования на ЭВМ различных явлений. Ведь познавательной целью является вовсе не изучение результата деятельности (в отрыве от процесса) и не изучение самого процесса (в отрыве от результата). Главное здесь — именно взаимное соответствие структуры деятельности и структуры результата этой деятельности! Важно отметить, что решение этой двухаспектной задачи в конечном счете представляется ученикам гораздо более простым и естественным, чем занятия толь-

ко процедурным (алгоритмическим) аспектом всякой деятельности.

Что же представляет собой с этой точки зрения интерпретатор языка Пролог? Это «игровая» среда, в рамках которой по несложным правилам (а синтаксис языка Пролог очень прост) создаются «базы знаний». При этом предметные области знаний могут быть самыми разными; их выбор зависит от интересов учеников. База знаний, с одной стороны, представляет собой декларативное описание (т. е. описание структуры объекта, а вместе с запросом к базе знаний — описание цели деятельности). С другой стороны, ученики с самого первого занятия по Прологу овладевают процедурной семантикой языка (т. е. описанием структуры деятельности). Тем самым ученики получают возможность изучать, как те или иные изменения в целеуказании влияют на изменение способа достижения поставленной цели; что общего должно быть у двух различных процедур, чтобы обе они приводили к достижению одной и той же цели и т. д.

Из вышеизложенного следует, что целью учебной информатики не может служить решение какой-либо частной дидактической проблемы — формирования «алгоритмического мышления», например. С научной точки зрения методологической целью предмета может являться обучение исследованию соотношения структуры цели и структуры деятельности по достижению данной цели. Конкретно, какие именно компоненты цели деятельности определяют те или иные элементы самой деятельности.

Поэтому необходимо, занимаясь Прологом, не забывать и о процедурах (алгоритмах!) поиска ответов на вопросы. Конечно, того же эффекта можно достичь, занимаясь алгоритмизацией с обязательным анализом соответствия каждого алгоритма предполагаемому результату его выполнения. Естественно, это требует четкого описания результатов выполнения каждого алгоритма. Учитывая, однако, что описание результатов выполнения алгоритма требует применения достаточно сложных для школьников формализмов, более предпочтительным представляется подход «от Пролога».

Итак, главное предостережение всем начинающим использовать язык Пролог в обучении: не забывайте о процедурах! Пускай ученики устанут от ваших требований объяснить, как именно ЭВМ будет искать ответ на каждый конкретный вопрос, адресованный базе знаний. Только при этом условии использование декларативного языка поможет добиться дидактического эффекта, недостижимого при помощи алгоритмизации и процедурного программирования.

Далее предлагаем методические материалы по языку Пролог, призванные помочь учителям (а также всем интересующимся) познакомиться с этим языком логического программирования. Последовательность изложения напрямую не связана с каким-либо вариантом преподавания курса информатики, однако надеемся, что публикация даст возможность «погрузиться» в достаточно своеобразный мир этого языка. Наряду с описанием основных возможностей языка Пролог излагаются также необходимые элементы логики.

Е. ЕРОХИНА
(Москва)

От логики к программированию (Пролог в курсе информатики)

В данной статье для написания программ используется интерпретатор языка Пролог-Д, созданный под руководством С. Г. Григорьева. Интерпретаторы этого языка разработаны для компьютеров БК-0010, «Ямаха», УКНЦ, IBM PC.

После каждого раздела приведены задачи и упражнения, предназначенные для закрепления материала. Ответы и решения к ним приведены в конце статьи.

1. Поговорим о языке

Все мы с детства пользуемся различными языками и не видим в этом ничего сложного. Если посмотреть на все языки (в том числе и языки программирования), можно обнаружить, что все они состоят из трех основных «частей» (составляющих):

/ Алфавит
ЯЗЫК: — Синтаксис
 \ Семантика

Дадим определения:

Алфавит — это набор символов, используемых в языке.

Например,

1. Русский алфавит:

А Б В Г Д Е Ж З И Й К Л М
 Н О П Р С Т У Ф Х Ц Ч Ш Щ

ь Ы Ь Э Ю Я
 а б в г д е х з и й к л м
 н о п р с т у ф х ц ч ш щ
 ь ы ь э ю я
 . , : - ! ? ; < >

2. Алфавит десятичных чисел:

0 1 2 3 4 5 6 7 8 9 , + -

3. Алфавит двоичных чисел:

0 1 ,

4. Алфавит азбуки Морзе:

. - < >

< > — пробел (единичное расстояние между словами).

Синтаксис определяет правила построения конструкций языка из символов алфавита.

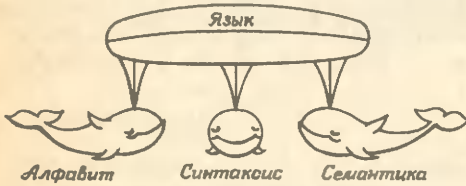
Конструкциями русского языка можно считать слова и предложения, языка математики — числа и формулы, а для алгоритмических (процедурных) языков программирования конструкциями являются операторы языка. Далее мы для простоты часто будем называть конструкции языков «словами». В большинстве языков из символов алфавита в принципе можно получить бесконечное множество слов. Но это верно не для всех языков. Ниже будет рассмот-

рен пример языка, в котором множество слов конечно.

Наконец, третий «кит» языка:

Семантика — система правил истолкования конструкций языка.

Для языков программирования семантика означает правила истолкования операторов.



Например, рассмотрим фрагмент оператора:

... $a * (x + b)$...

58

Правила семантики языка Паскаль предписывают вначале вычислить сумму, а затем умножить результат на «а». Однако можно

2. Формальное описание языка

Синтаксис различных языков, в том числе и языков программирования, удобно

$\langle \dots \rangle$ — "об'ясняемые" понятия |
 заключаются в угловые |
 скобки |
 ::= — это один символ; чита- |
 ется как "это", "есть", |
 "определяется как" |
 | — разделяет разные вари- |
 анты "об'яснения" |

Например, попробуем описать при помощи БНФ синтаксис языка неотрицательных целых двоичных чисел. Двоичным числом

$\langle \text{цифра} \rangle ::= 0 \mid 1$ |
 $\langle \text{число} \rangle ::= 1 \mid \langle \text{число} \rangle \langle \text{цифра} \rangle$ |

Можно также описать язык десятичных чисел:

$\langle \text{десятичное число} \rangle ::= \langle \text{число} \rangle \mid \langle \text{знак} \rangle \langle \text{число} \rangle$ |
 Десятичные числа могут |
 иметь знак или не |
 иметь знака. |

придумать язык, в котором семантика этого фрагмента будет иной.

Задачи. 1. Найти и исправить ошибки, указав, к какому классу их можно отнести (ошибки алфавита, синтаксические, семантические):

1. В песу родилась 2*2.
2. MAN EAT THE TABLE.
3. Сумма этих 2 чисел = 3.
4. $15:3 = 4$
5. @Карова впадает в кас- пийское море.
6. $1 + 1 = 10$

описывать с помощью нормальных форм Бэкуса — Наура (сокращенно БНФ).

В БНФ используются следующие символы:

Пример:

$\langle \text{предложение} \rangle, \langle \text{число} \rangle$ |
 $\langle \text{цифра} \rangle, \langle \text{понятие} \rangle.$ |
 |
 |
 $\langle \text{цифра} \rangle ::= 0$ |
 Читается: |
 Цифра — это 0. |
 |
 $\langle \text{буква} \rangle ::= a \mid b$ |
 Читается: |
 Буква — это a или b. |

будем считать последовательность нулей и единиц, начинающуюся с единицы.

| Цифра — это 0 или 1. |
 | 1 — это двоичное число. |
 | Если к двоичному числу |
 | сзади приписать цифру, |
 | также получится двоич- |
 | ное число. |

<знак> ::= + | -

<число> ::= <целое>
| <целое>, <целое>

<целое> ::= <цифра>
| <целое><цифра>

<цифра> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0

| Знак - это плюс либо
| минус.

| Числа бывают целыми,
| но могут иметь и
| дробную часть.

| Целое содержит одну
| или несколько цифр.

| В десятичной системе
| счисления используются
| цифры от 0 до 9.

Все предложения, записанные в форме БНФ, можно прочитать по-русски. Соответствующие фразы приведены в правой колонке.

Четвертое правило требует особого разъяснения. Это правило гласит, что целое число состоит из цифр — одной или нескольких. Число, состоящее из одной цифры, образуется при помощи первой части правила. Число, в котором более одной цифры, может быть образовано путем последовательного применения обеих частей четвертого правила. Вначале получается первая цифра числа (по первой части правила). Затем к этой цифре справа приписываются все остальные цифры (это позволяет делать вторая часть правила). Иными словами, любое правило, входящее в БНФ, можно применять несколько раз.

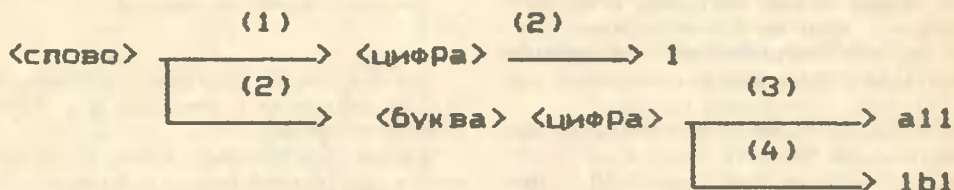
Например, опишем синтаксис языка, алфавит которого состоит из символов (a, b, k, 2). Для удобства правила занумеруем.

- (1) <слово> ::= a
- (2) <слово> ::= <слово> b
- (3) <слово> ::= <слово> k
- (4) <слово> ::= 2 <слово>

Выясним, какие слова входят в этот язык и как их можно получить.

1) a — очевидно, это слово можно получить, применив правило (1).

2) abbk — укажем последовательность,



Мы видим, что все возможные способы применения правил исчерпаны, и следовательно, мы получили все слова этого языка.

Чем же отличаются описания синтаксиса языка с бесконечным и конечным количеством слов? В языке с конечным количе-

в которой надо применять правила, чтобы получить это слово:

a ak akb 2akb
(1) → (3) → (2) → (4) →

Здесь в скобках записаны номера правил, а над стрелками показаны результаты их применения.

3) 2akb — получается аналогичным образом:

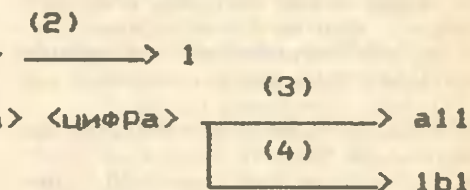
a ab abb abbk
(1) → (2) → (2) → (3) →

4) 2a2k — получить нельзя, так как цифра 2 по описанию синтаксиса может приписываться к слову только слева, а все буквы (кроме a) — только справа, и следовательно, двойка не может стоять в середине слова.

Этот язык содержит бесконечное число слов. А теперь, как было обещано выше, рассмотрим язык, число слов в котором конечно.

- (1) <слово> ::= <цифра>
| <буква> <цифра>
- (2) <цифра> ::= 1
- (3) <буква> ::= a <цифра>
- (4) <буква> ::= <цифра> b

Изобразим на схеме все возможные способы применения этих правил:



ством слов понятия определяются только через другие понятия. Для языка с бесконечным количеством слов при описании понятия может использоваться это же понятие (определение «через само себя»). (Напри-

мер, в правиле 4 для языка десятичных чисел описание целого числа содержит понятие «целое»).

Задачи и упражнения. 1. Описать в БНФ:

- а) язык шестнадцатеричных чисел;
- б) язык, в состав которого входят слова: a ; $a^?$; $a!$; $a-a$; $a-a-a?$; $a-a!$ и т. п.;
- в) язык, фразы которого строятся из следующих составляющих: ну, а, стало быть, так сказать, примерно, конечно, вот, э, э-э, э-э-и т. п. и предположениями которого являются произвольные сочетания этих слов. Например: ну вот. стало быть ну э-э конечно. так сказать примерно. В конце каждого предложения должна стоять точка.

2. Дано описание языка в форме БНФ.

- (1) $\langle \text{слово} \rangle ::= Q$
- (2) $\langle \text{слово} \rangle ::= \langle \text{кусок} \rangle$
- (3) $\langle \text{слово} \rangle ::= \langle \text{кусок} \rangle \langle \text{кусок} \rangle$
- (4) $\langle \text{слово} \rangle ::= \langle \text{слово} \rangle 2$
- (5) $\langle \text{слово} \rangle ::= \langle \text{слово} \rangle 7$
- (6) $\langle \text{кусок} \rangle ::= a$
- (7) $\langle \text{кусок} \rangle ::= 7$
- (8) $\langle \text{кусок} \rangle ::= c \langle \text{слово} \rangle c$

Выяснить, выводимы ли слова:

- а) $sa7c$ в) $saQ77c$ д) $77c77a$
- б) $QQ27a$ г) $ссacc$ е) $сxQc$

Указать, в какой последовательности надо применять правила для получения этих слов.

3. Немного логики

Под суждением мы будем понимать утверждение, истинность которого (в общем случае) зависит от значений входящих в него переменных.

Пусть $A(X)$ и $B(Y)$ — суждения. По правилам логики их можно объединять при помощи логических связок. Для каждой логической связки можно составить истинностную таблицу, выписав все возможные сочетания значений переменных, соединенных связкой. Ниже приведены истинностные таблицы основных логических связок.

1. Связка не (отрицание, инверсия; обозначается \neg или $-$).

$\text{не } A$ обозначает отрицание A . Для того чтобы получить отрицание суждения A , надо начать фразу-отрицание со слов «Неверно, что...» и далее записать само суждение A . Например:

Суждение:
« $a+b$ равно 0.»

Отрицание:
«Неверно, что $a+b$ равно 0.»

«В понедельник был дождь»

«Неверно, что в понедельник был дождь» (то есть в понедельник дождя не было).
«Неверно, что эта теорема неверна» (то есть теорема верна).

«Эта теорема неверна».

Отрицание можно записывать и в другой форме:

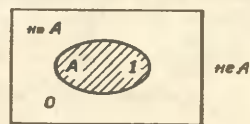
$\text{НЕ } \{ (a < b) \} \Leftrightarrow \{ a \geq b \}$
("не меньше" — то же, что "больше или равно").

(«не меньше» — то же, что «больше или равно»).

$\text{НЕ } (\text{«В Африке живут крокодилы»}) \Leftrightarrow (\text{«В Африке не живут крокодилы»})$

Запишем истинностную таблицу:

A	не A
1	0
0	1

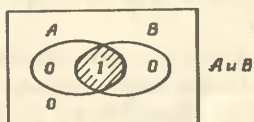


Здесь единицей обозначено истинное (при данных значениях переменных), а нулем — ложное суждение.

Схема, приведенная слева (она называется диаграммой Венна), обозначает множество всех возможных значений параметров (X) . Круг — множество значений X , при которых истинно A . Область вне круга — подмножество X , где истинно не A . Из диаграммы видно, что не A истинно тогда и только тогда, когда ложно A .

2. Связка и (конъюнкция, или «логическое умножение»; обозначается \wedge или $\&$).

A	B	A и B
0	0	0
0	1	0
1	0	0
1	1	1



Пересечение кругов на диаграмме есть множество тех и только тех элементов X, при которых одновременно истинны A(X) и B(X).

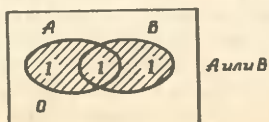
Высказывание A и B истинно тогда и только тогда, когда истинны оба составляющих его высказывания.

Связка и иногда называется «логическим умножением», так как ее значения совпадают с результатами умножения A на B. (Действительно,

$$\begin{array}{l}
 1 * 1 = 1 \quad \text{и} \quad 1 \& 1 = 1, \\
 1 * 0 = 0 \quad \text{и} \quad 1 \& 0 = 0, \\
 0 * 1 = 0 \quad \text{и} \quad 0 \& 1 = 0, \\
 0 * 0 = 0 \quad \text{и} \quad 0 \& 0 = 0
 \end{array}$$

3. Связка или (дизъюнкция, или «логическое сложение»; обозначается знаком \vee).

A	B	A или B
0	0	0
0	1	1
1	0	1
1	1	1



Множество истинности A или B — объединение кругов, обозначающих множества, где истинны A и B соответственно.

Связка или истинна тогда, когда истинно хотя бы одно из соединенных ею высказываний.

Название «логическое сложение» для связки или не совсем точно, так как результат сложения A и B совпадает со значением этой связки только в 3 случаях: $0+0=0$, $0+1=1$, $1+0=1$. В случае, когда $A=1$ и $B=1$, имеем: $A \vee B=1$, $A+B=10$ (если рассматривать запись результата сложения в двоичной системе счисления). Таким образом, правильнее было бы назвать связку или «аналогом двоичного сложения с отбрасыванием переноса».

Связка или в житейских суждениях может играть объединяющую либо разделяющую роль. Например, «Снег пойдет сегодня или завтра» означает, что снег может пойти сегодня, может — завтра, но может идти и оба дня подряд.

Однако роль связки или может быть и разделяющей: например, в утверждении «Мы поедем поездом или полетим самолетом». Там, где связка или играет разделяющую роль, ее можно заменить словом «либо» (смысл утверждения при этом не изменится).

В логике связка или всегда играет объединяющую роль.

Задачи и упражнения: 1. Верно ли, что:

- НЕ («Москва — столица России») = («Владивосток — столица России»);
- из двух противоречивых высказываний ровно одно истинно;
- из двух противоречивых высказываний хотя бы одно ложно.

Противоречивыми высказываниями будем считать пару высказываний, которые не могут быть истинными одновременно.

2. Составьте таблицу истинности:

- не (a и (b или c));
- не (a или b);
- (a и b) или не c;
- b или не (a и c).

4. «Может ли машина мыслить?», или Начинаем изучать Пролог

Часто можно услышать отрицательный ответ на вопрос: «Может ли машина мыслить?» Прежде чем согласиться с этим мнением или с ним спорить, следует определить, что понимается под способностью к мышлению. Если свести процесс мышления к проведению осмысленных рассуждений, то можно считать, что машина способна «мыслить», если ей сообщить исходные факты и правила вывода умозаключений. После того как это сделано, ей можно задавать вопро-

сы, на которые ЭВМ будет давать осмысленные ответы.

Определим основные понятия.

Базой данных назовем набор сведений, записанных в форме, «понятной» ЭВМ.

Базой знаний будем называть базу данных, в которой можно накапливать информацию в памяти ЭВМ, искать нужную информацию и получать от машины логически осмысленные заключения.

Для создания баз знаний используются языки логического программирования, к числу которых относится язык Пролог.

База знаний на Прологе может содержать предложения трех типов:

- факты;
- правила;
- вопросы.

Факты — это сведения об интересующем нас явлении, записанные на языке, который «понимает» ЭВМ.

Например, мы хотим «сообщить» компьютеру сведения о семье: «Наталья — мама Марии»; «Анна — мама Натальи».



Прежде всего надо выбрать форму для записи фактов о мамах. Будем записывать эти факты в виде:

мама (<кто>, <чья>)

«> обозначает конец факта (и вообще любого предложения) на Прологе-Д.

Наши сведения в форме, «понятной» ЭВМ, будут выглядеть так:

мама (Наталья,Мария);
мама (Анна,Наталья);

Порядок следования объектов задается один раз (в начале построения базы знаний), а затем мы должны будем придерживаться его при записи всех фактов (а также при создании правил и при постановке вопросов).

Введя эти факты в ЭВМ, мы сможем задавать машине вопросы. Далее мы будем записывать вопросы и ответы в трех колонках: в первой — вопрос на русском языке, во второй — этот же вопрос на Прологе, а в третьей — ответ ЭВМ.

62

По-русски:	На Прологе:	Ответ ЭВМ:
Как зовут маму Марии?	?мама(х,Мария);	х=Наталья
В вопросах на месте неизвестного объекта ставится имя переменной (одна русская или латинская буква). Обратите внимание, что	все имена в фактах и вопросах записываются в именительном падеже.	Зададим еще несколько вопросов:
По-русски:	На Прологе:	Ответ ЭВМ:
Чья мама Анна?	?мама(Анна,γ);	γ=Наталья
Кто чья мама?	?мама(х,γ);	х=Наталья γ=Мария х=Анна γ=Наталья
Кто мама γ мамы Марии?	?мама(х,Мария), мама(γ,х);	х=Наталья γ=Анна

Запятая в вопросе обозначает логическую связь и.

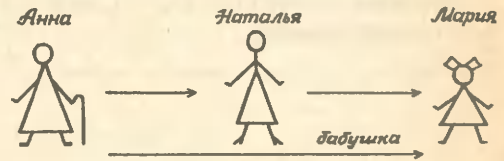
С точки зрения человека, в нашей базе достаточно сведений для того, чтобы сказать,

как зовут бабушку Марии. Однако если напрямую спросить об этом машину, мы получим ответ «НЕТ»:

По-русски:	На Прологе:	Ответ ЗВМ:
"Кто бабушка Марии?"	?бабушка(x, Мария);	НЕТ

Это вовсе не означает, что в базе недостаточно информации для того, чтобы найти имя бабушки. Пролог просто не знает «определения» бабушки.

Запишем правило для «определения» бабушки в форме:



бабушка (<кто>, <чья>)

По-русски:	На Прологе:
Бабушка — это мама мамы.	бабушка(x, y) ←-мама(x, z), мама(z, y);

Знак «←» читается как «если», «это», «когда». получить ответ на вопрос: «Кто бабушка Марии?»:

63

Введя это правило в машину, мы сможем

По-русски:	На Прологе:	Ответ ЗВМ:
"Кто бабушка Марии?"	?бабушка(x, Мария);	x=Анна

По этому правилу машина сделала логический вывод о факте, которого в исходной базе данных нет. Мы «научили» машину рассуждать, сообщив ей для этого правила, по которым проводятся рассуждения.

Пролог всегда ищет ответ на поставленный вопрос по одному и тому же алгоритму. Сначала он пробует найти факт, при помощи которого можно ответить. Если такого факта нет, машина начинает применять правила. Для ответа на один вопрос может применяться несколько правил.

Для того чтобы получить логическую связку или, надо факты и правила, связанные ею, перечислить по отдельности.

Задачи. Написать на Прологе правила для определения:

- дедушки по материнской линии (дополнив базу знаний сведениями о папах);
- бабушки и дедушки по отцовской линии.

5. «Кантор придумал кванторы»

Рассмотрим два определения.

Остроугольный треугольник — тот, у которого все углы острые (т. е. величины всех углов меньше 90 градусов). (1)

Тупоугольный треугольник — тот, у которого есть тупой угол (т. е. угол, величина которого больше 90 градусов). (2)

Принципиальная разница в структуре этих определений заключается в том, что в определении (1) используется квантор \forall все-

общности (читается «для всех» или «для каждого»). Обозначение впервые введено Кантором и происходит от слова Alles.

Иногда этот квантор читается как «для любого». Это следует признать неудачным, так как слово «любой» в зависимости от контекста может иметь значение «каждый» или «существует».

В определении (2) используется квантор существования (обозначается \exists , читается «существует», «найдется», «можно подобрать»). Он происходит от слова Exists.

Пусть $P(x)$ — некоторое суждение.

Тогда $\exists x:P(x) \equiv \forall x:\overline{P(x)}$. Иначе говоря,

НЕ (существует X такое, что истинно $P(X)$) = для каждого X НЕ ($P(X)$).

То есть

Для отрицания существования необходимо доказать ложность утверждаемого во всех случаях

Аналогично $\overline{\forall x:P(x)} \equiv \exists x:\overline{P(x)}$, то есть существует X такое, что НЕ ($P(X)$).

Для отрицания общего утверждения достаточно привести хотя бы один контрпример.

Рассмотрим в качестве примера базу знаний из школьной жизни. Пусть в базе имеются факты вида:

оценка (<Фамилия>, <Предмет>, <Балл>)

Например, база может содержать следующий набор фактов об оценках:

оценка (Иванова, литература, 5);
 оценка (Иванова, математика, 5);
 оценка (Иванова, музыка, 5);
 оценка (Петров, физика, 4);
 оценка (Петров, химия, 3);
 оценка (Петров, литература, 5);
 оценка (Васечкин, музыка, 4);
 оценка (Васечкин, физика, 5);
 оценка (Васечкин, литература, 2);
 оценка (Васечкин, литература, 5);

Введя эту базу, можно задать вопросы об оценках, например:

По-русски:	На Прологе:	Ответ ЭВМ:
Какая оценка по литературе у Ивановой?	?оценка(Иванова, литература, x);	x=5
У кого есть тройка по химии?	?оценка(a, химия, 3);	a=Петров
У кого есть пятерки?	?оценка(x, _, 5);	x=Иванова x=Иванова x=Иванова x=Петров x=Васечкин x=Васечкин

Если у одного и того же ученика несколько пятерок, ЭВМ повторит его фамилию столько раз, сколько таких фактов содержится в базе знаний.

Символ «—» в третьем вопросе означает, что переменная, на месте которой он стоит,

может принимать какое угодно значение (и нас оно не интересует). Если же мы хотим в ответе на вопрос увидеть и предметы, по которым стоит та оценка, которая нам нужна, то «на месте» предмета надо поставить еще одну переменную, например:

По-русски:	На Прологе:	Ответ ЗВМ:
У кого есть четверки и по какому предмету?	?оценка (x, y, 4);	x=Петров y=физика x=Васечкин y=музыка

Дадим определение двоечника (в этом правиле будет один параметр: фамилия двоечника):

По-русски:	На Прологе:
Двоечник — тот, у кого есть хотя бы одна двойка по какому-либо предмету.	двоечник (x) ← оценка (x, _, 2);

Введя это правило в машину и задав вопрос, получим ответ, что двоечник — Васечкин. Попробуем определить «физика» и «лири-

ка»: «Лирик — тот, у кого пятерка по литературе, а физик — тот, кто не лирик (и имеет хотя бы одну оценку по физике).

65

По-русски:	На Прологе:
Лирик — тот, у кого пятерка по литературе. Физик — тот, кто имеет хотя бы одну оценку по физике и не лирик.	лирик (x) ← оценка (x, литература, 5); физик (x) ← оценка (x, физика, _), НЕ (лирик (x));

«НЕ» во втором правиле обозначает логическую связку не (отрицание).

А как дать определение отличника? Попробуем дать такое определение:

отличник (x) ← оценка (x, _, 5);

Задав машине соответствующий вопрос (?отличник(x));, в ответ получим фамилии предполагаемых отличников: Иванова, Петров, Васечкин. Таким образом, машина «счи-

тает» отличниками всех, у кого есть пятерки (а по школьным правилам отличником считается тот, у кого нет других оценок, кроме пятерок). Как же быть?

Попробуем вместо отличника описать того, у кого есть другие оценки (кроме пятерок). Для этого запишем вспомогательное правило:

По-русски:	На Прологе:
Тот, у кого есть хотя бы одна оценка ниже пяти — не отличник.	неотличник (x) ← оценка (x, _, a), МЕНЬШЕ (a, 5);

Встроенная функция МЕНЬШЕ служит для сравнения двух величин, указанных в качестве аргументов.

Тогда правило для определения отличника будет выглядеть так:

По-русски:	На Прологе:
Отличник — тот, у кого есть хотя бы одна пятерка и кто не является неотличником.	отличник (x) ← (x, _, 5), НЕ (неотличник (x));

Примечание. В именах фактов и правил можно использовать символ « » (подчеркивание), например «не блестяще». Тире, точки и другие специальные знаки в именах использовать не разрешается.

Задача. Пусть в базе знаний «школа», кроме сведений об оценках, содержатся факты вида:

собирает (<фамилия>, <что>)

Записать на Прологе определения:

- а) филателист — тот, кто собирает марки;
- б) меломан — тот, кто собирает ноты или пластинки;

в) книголюб — тот, кто собирает книги и имеет оценку 4 или 5 по литературе;

г) музыкант — меломан, который имеет одни лишь пятерки по музыке.

7. Правила де Моргана

В математической логике есть два правила, которые применяются при получении отрицания конъюнкции и дизъюнкции. Они называются правилами де Моргана.

1. Отрицание конъюнкции:

$$\text{не } (A \text{ и } B) \Leftrightarrow (\text{не } A) \text{ или } (\text{не } B)$$

Здесь знак \Leftrightarrow обозначает равносильность (читается «тогда и только тогда, когда»).

Иными словами, отрицание конъюнкции

эквивалентно дизъюнкции отрицаний. Можно легко убедиться в справедливости этого правила, если построить таблицы истинности для левой и правой частей:

A	B	не (A и B)	(не A) или (не B)
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Пример:

$$\text{НЕ } ((3 \leq x \leq 5)) =$$

$$\text{НЕ } ((x \geq 3) \text{ и } (x \leq 5)) =$$

$$\text{НЕ } ((x \geq 3)) \text{ или}$$

$$\text{НЕ } ((x \leq 5)) =$$

$$(x < 3) \text{ или } (x > 5)$$

2. Отрицание дизъюнкции:

$$\text{не } (A \text{ или } B) \Leftrightarrow (\text{не } A) \text{ и } (\text{не } B)$$

Это правило также можно проверить, составив таблицы истинности для его левой и правой частей.

Правила де Моргана часто используются в программировании для записи различных комбинаций логических связей. Мы будем использовать эти правила при написании баз знаний на языке Пролог.

8. Пролог: логика и математика

Мы уже познакомились с употреблением логических связок. Обобщим сведения, которые нам известны:

Для реализации связки не используется встроенная функция НЕ (отрицание) в правой части правил и в вопросах.

Для реализации связки и используется запятая в правой части правила или в вопросе.

Связка или на Прологе не имеет специального обозначения. Для того, чтобы реализовать эту связку, записывается несколько правил с одинаковой левой частью и различными правыми, или несколько вопросов.

Кроме того, в Прологе есть встроенные функции для сравнения двух величин. (В предыдущем параграфе мы уже использовали функцию МЕНЬШЕ). Ниже приведена таблица истинности для этих функций.

Функция	"Истина", если
МЕНЬШЕ (a, b);	a < b
БОЛЬШЕ (a, b);	a > b
РАВНО (a, b);	a = b

Квантор существования на Прологе обозначается символом «_» в правой части правила. Для квантора всеобщности специального обозначения на Прологе нет, но его можно заменить квантором существования, используя правила для отрицания, похожие на правила де Моргана для логических связок:

НЕ (существует X, такое что истинно P(X)) <=> для каждого X истинно НЕ (P(X)).

НЕ (для каждого X истинно P(X)) <=> существует X, такое что истинно НЕ(P(X)).

Рассмотрим использование этих правил на примере базы знаний «Гидрометцентр».

Пусть наблюдения за погодой за неделю дали следующие результаты:

День недели	ПН	ВТ	СР	ЧТ	ПТ	СБ	ВС
Параметры							
Температура (С)	15	12	7	0	5	10	20
Давление (мм.рт.ст.)	760	754	766	700	780	750	760
Направление ветра	юж	сев	вост	сев_в	ю_зап	ю_в	юж

(Продолжение следует)

Доска объявлений

Ищу единомышленников, владельцев компьютера «Спектрум». Могу предложить большое количество программ, в том числе и абсолютно новые, как системные, так и игровые, фирменные и отечественного производства. Имею также русифицированный «Спектрум» и принтер. Высылаю каталог.

330118, г. Запорожье, ул. Автозаводская, д. 20, кв. 221.

В. Г. Печерский

Приглашаю владельцев БК-0010(01) и «Спектрум» обмениваться программами, опытом и идеями. Высылаю каталог. Могу помочь с оформлением игр: заставки, музыка, спрайты.

716016, Узбекистан, г. Наманган, 4 микрорайон, д. 53, кв. 21.

Э. Л. Гемза

Приобрету кассету с играми для компьютера «Электроника КР02». Возраст играющего — 12—13 лет.

430023, г. Саранск, пр. 60 лет Октября, д. 10, кв. 117.

О. А. Тешина

Имею большую коллекцию программ для БК-0010-01 и ряд практических пособий для желающих научиться программировать в кодах. Занимаюсь адаптацией программ для ДООС БК.

394000, г. Воронеж, а/я 236.

Г. В. Черкасов

Предлагаю организовать заочный клуб владельцев «Поиска».

393350, г. Кирсанов Тамбовской обл., ул. Советская, д. 36, кв. 5.

А. Н. Патрин

Имею около 2000 программ для БК-0010 и около 1000 для других компьютеров. Высылаю каталог. Могу помочь в создании спрайтов для БК-0010.01.

446800, с. Кошки Самарской обл., квартал 2, д. 3, кв. 6.

А. Е. Федоров

Прошу откликнуться тех, кто может помочь с программами для ПК-8000 («Сур», «Веста» и др.).

626726, г. Ноябрьск-7 Тюменской обл., просп. Мира, д. 30, кв. 52.

А. В. Дмитриев

Обменяюсь программами для БК-0010. 445023, г. Тольятти-23, до востребования.

А. Ю. Слюсарев

Предлагаю учителям информатики, работающим с классами УКНЦ, переписываться и обмениваться программами.

288387, Украина, с. Черномин Песчанского р-на Винницкой обл., ул. Пушкина, д. 11.

Р. Н. Броль

Учитель, «разлученный» с «Корветом», меняет весь комплект ПС на «Корвет» (DOS, CP/M).

659540, Алтай, с. Турочак, ул. Воробьевского, 2А-1.

И. С. Истомин

Хочу переписываться с пользователями ПЭВМ «Агат-7,9», программирующими на Ассемблере. Возможен обмен ПО (ЕС 5323.01).

310028, г. Харьков, ул. С. Тархова, д. 1, кв. 33.

Б. Иванов

Программы для компьютеров: БК-0010.01, 11М (на кассетах и дискетах 5.25); ZX SINCLAIR SPECTRUM и совместимых с ним (на кассетах); ATAPI XE, XL (на кассетах и дискетах 5.25); ATARI ST (на дискетах 3,5); COMMODORE 64/128 (на кассетах и дискетах 5.25); AMIGA (на дискетах 3,5); IBM PC и совместимых с ним (на дискетах 5.25 и 3.5) по адресу: 127349, г. Москва, а/я 9. Тел. (095) 908-22-12 ежедневно с 10 до 21 часа.

В. П. Юров

Помогите, пожалуйста, начинающему пользователю БК-0010 советом, программным обеспечением. У меня нет ни монитора, ни магнитофона. Буду очень благодарен.

627453, с. Окунево Бердюжского р-на Тюменской обл.

С. Замарев

Предлагаю всем владельцам «Вектора 06 Ц» программу «Клавиатура» — клавиатурный тренажер, обучающий десятипальцевой системе набора (вслепую), а также проверяющий производительность в зн/мин.

Возможен обмен программами. Вышлите кассету по адресу:

270550, Молдова, г. Единцы, ул. Дзержинского, 21/2.

А. Е. Пашинский

Владельцам БК-0010.01 предлагаю переписку и обмен программами. Имею более 600 программ, вышло каталог.

659430, Алтайский край, с. Целинное, а/я 1.

А. В. Коротеев

Владельцам «Микроши» предлагаю обмен программами. Имею в распоряжении более 200 программ для этого компьютера. Готов выслать всем учителям ОИВТ сборник методических материалов и задач по курсу 10—11 классов.

424004, г. Йошкар-Ола, ул. Маяковского, д. 3.

Е. Н. Смирнов

Хочу переписываться с владельцами ПК «Синклер» (желательно наличие дисковода и операционной системы CP/M). Имею на кассетах около 200 программ. Программирую на Бейсике, Турбо-Паскале, немного на dBase, хотел бы научиться программировать в машинных кодах «Спектрума».

220118, г. Минск, ул. Кабушкина, д. 92, кв. 34.

М. И. Корнейчик

Хочу переписываться с владельцами компьютера «Вектор 06Ц». Имею более 30 системных и игровых программ. Пишу их сам.

УВАЖАЕМЫЕ ЧИТАТЕЛИ!

В частном объявлении стоимость одного знака, включая пробел, 20 руб. Адрес в стоимость не входит. Приведенная квитанция содержит банковские реквизиты для жителей России (кроме Москвы и Московской области). Копию платежного поручения и текст объявления высылайте в редакцию. Тех, кто живет в Москве и Московской области, приглашаем приносить свои объявления в редакцию по адресу: Москва, ул. Садово-Сухаревская, д.16, комн.9.

Есть идея создать клуб пользователей этого компьютера.

462430, г. Орск Оренбургской обл., ул. Добровольского, д. 13, кв. 70.

Е. В. Черников

Обменяюсь дисковыми версиями программного обеспечения для ПК «ZX — SPECTRUM» и любой другой информацией.

170033, г. Тверь, пр. Волоколамский, д. 16, кв. 68.

С. Б. Запруднов

Хочу переписываться и обмениваться программами с пользователями БК-0010.01 и «ZX — SPECTRUM».

686417, п. Дукал Омсукчанского р-на Магаданской обл., ул. Московская, д. 8, кв. 17.

В. С. Коваленко

Предлагаю помощь пользователям IBM PC, «Поиска», «Агата», БК-0010.01, «Атари», «Спектрума», «Коммодора», УКНЦ, ДВК, «Партнера», «Львова», ПК-800, «Правца», РК-86, «Специалиста», «Корвета» и других.

349010, Украина, п. Славяносербск Луганской обл., а/я 34.

Р. И. Жага

Для БК-0010 (01) игры в кодах: новейшая «СПИД» и управляемая голосом «Т-игра». Направляйте кассету по адресу: 155313 Ивановская обл., Вичугский р., п. Новописцово, Заречная, б. *Воронову П. Н.*

КВИТАНЦИЯ Кассир	Государственное предприятие «Редакция журнала «Информатика и образование» ММКБ Филиал «Интеллект» р/с 609602			
	Получатель платежа			
	Учреждение банка	ЦРКЦ ГУ ЦБ РФ по М.о. МФО 211004		
	Счет получателя	216161800		
	Вид платежа			Сумма
	За публикацию объявления			

„Завуч“

© Фирма «Колледж» (095)267-70-55

Система построения
расписания занятий

(УК-НЦ - новая версия 2.0,
ДВК, IBM - версия 1.3)

„Мини-канцелярия“

© Фирма «Колледж» (095)267-70-55

Автоматизированная
система для подготовки,
хранения, поиска и ка-
чественной печати
документов
(УК-НЦ).

„Бухгалтерия“

© Фирма «Колледж» (095)267-70-55

Автоматизированная
система бухгалтерского
учета (УК-НЦ).

„Склад“

© Фирма «Колледж» (095)267-70-55

Автоматизированная
система складского
учета (УК-НЦ).

КОЛЛЕДЖ

Пакет административных программ

Наш
адрес:
107005,
Москва
Волховский
пер., 11
Фирма «Колледж»
телефон: 267-7058

Пользователям СПР „Завуч“ версии 1.3 предоставляется скидка при приобретении версии 2.0



РАЗЛОЖИМ ЗВУК, ВЗВЕСИМ И ИСЧИСЛИМ

71

Д. УСЕНКОВ
(Москва)

Музыкальный редактор для Бейсик-Вильнюса БК-0010.01

Внимательный пользователь БК-0010.01 легко может заметить, что зашитая в его ПЗУ Вильнюсская версия Бейсика очень напоминает широко распространенную версию MSX. Однако Бейсик-БК, под который создатели БК-0010 «щедро» отвели всего 20 кбайт «с хвостиком», в отличие от стандарта MSX заметно обделен возможностями. В Бейсике БК-0010 не реализован режим «отлова» прерываний клавиатуры в Бейсик-программе (оператор MSX-Бейсика KEY(<n>)), нет возможности отлавливать в самой Бейсик-программе возникшие при ее работе ошибки (ON ERROR GOTO...), не предусмотрено программное обслуживание джойстика и таймера да и вообще порта ввода-вывода, нет спрайтов, нет возможности вывода символов произвольных размеров, нет оператора PRINT USING, из-за чего не очень-то удобно выводить результаты расчетов на экран или на принтер в виде таблиц, и, что особенно скверно, совсем нет звуковых эффектов. Поэтому у пользователей БК нет другого выхода, как только писать дополнительные сервисные USB-подпрограммы для Бейсика (хорошо хоть такая возможность в Бейсике есть, в отличие от Фокала!).

Предлагаемая программа предоставляет возможность обеспечить некоторое музыкальное сопровождение для программ на Бейсике. Конечно, БК с ее слабеньким пьезодинамиком не сравнится с оснащенными несколькими каналами «большими» машинами, но, если использовать магнитофон в качестве усилителя (установив его в режим записи и «утопив» клавишу «Пауза») и смириться с несколько «металлическим» звуком (это вызвано тем, что получаемые звуковые колебания имеют не синусоидальную, а скорее прямоугольную форму), результат получается вполне сносный.

Предлагаемая программа реализации оператора PLAY, конечно же, сильно упрощена по сравнению со стандартом MSX. В ней более условны обозначения нот (таблица кодирования близка к принятой в программе SYST3, упрощено задание длительности и, конечно же, нет многоголосности и возможности работы с волновыми пакетами. Но для такого упрощения есть и свои, довольно веские основания. Ведь наша программа должна будет находиться в ОЗУ вместе с программой на Бейсике, а резервы оператив-

ной памяти и так ограничены. Кроме того, программа — реализация PLAY должна быть «мобильной» и простой «в управлении», чтобы ее можно было без особых трудностей вставлять практически в любую программу на Бейсике.

Текст программы в машинных кодах необходимо ввести в БК с начального адреса &0360 0 0. О том, как это сделать, уже неоднократно писалось на страницах журналов по вычислительной технике. После того, как текст программы записан на магнитофон, ее можно использовать в Бейсике. Для этого нужно выполнить следующую последовательность действий:

```
CLEAR 200, &036000 ;
BLOAD "имя программы" ;
DEF USR<n>=&036000 (n — номер функции USR).
```

72

Теперь можно вызвать программу PLAY с помощью функции USR в виде: PRINT USR<n> («<текст, кодирующий последовательность нот>»), при этом текстовая константа или содержимое текстовой переменной после исполнения мелодии будет выведено на экран, или: АХ=USR<n> («<текст>»), где АХ — некоторая «буферная» текстовая переменная, значение которой в данный момент не требуется сохранять неизменным (такой прием позволяет не выводить текст на экран).

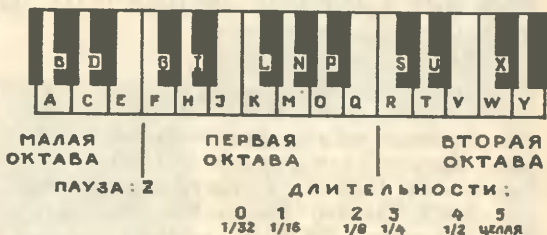
Следует отметить, что в случае вызова функции в виде АХ=USR(АХ), где АХ кодирует нотный текст, содержимое АХ может оказаться испорченным, поэтому такой способ вызова нужно применять с осторожностью.

Кстати, можно заодно дать пользователям БК такой немудреный совет. Чтобы всякий раз не подгружать коды подпрограммы PLAY (и прочие подпрограммы типа USR, а также спрайты и другие данные) перед запуском Бейсик-программы и одновременно избежать необходимости вставлять в нее множество операторов DATA с требуемыми кодами для того, чтобы сама Бейсик-программа заносила их в память операторами РОКЕ, можно, после того, как все требуемые USR-функции и кодовые данные загружены и «подключены» к программе, записать все вместе на магнитофон оператором BLOAD «<имя>», &020 0 0, &040 0 0. Такой файл занимает больше места, чем Бейсик-программа и кодовые «довески» к ней в отдельности умело применяя оператор CLEAR, иногда можно уменьшить размер записываемой области памяти, но зато сразу же после загрузки полученного таким образом. BIN-файла БК «автоматически» будет переведена в то же состоя-

ние, как и в момент записи вашей программы оператором BLOAD «<имя>», &020 0 0, &040 0 0 (вплоть до заданных оператором KEY значений программируемых ключей).

Немного о том, как кодировать мелодию. Текстовая строка, являющаяся аргументом функции USR, может содержать: латинские заглавные буквы от А до Z (обозначения нот и паузы), цифры от 0 до 5 (длительности) и точку <.> (задание длительности отдельной ноты в 1,5 раза большей, чем установлено для остальных нот). Прочие символы программой игнорируются. Длительность задается цифрой 0 — 5 по следующему правилу: 0 соответствует длительности 1/32, 1—1/16, ... 5 — целая. Конечно, в данном случае такое соответствие несколько условно, поэтому нужную длительность лучше всего установить подбором в каждом отдельном случае. Первым символом каждой «нотной» строки должна быть цифра (по умолчанию длительность устанавливается равной 0). Раз установленная

КОДИРОВАНИЕ НОТНОГО ТЕКСТА В ПРОГРАММЕ PLAY



ОКТАВА	НОТА	БУКВА
МАЛАЯ	соль# (ля&)	А
	ля	В
	ля# (си&)	С
	си	D
ПЕРВАЯ	до	F
	до# (ре&)	G
	ре	H
	ре# (ми&)	I
	ми	J
	фа	K
	фа# (соль&)	L
	соль	M
соль# (ля&)	N	
ля	O	
ля# (си&)	P	
си	Q	
ВТОРАЯ	до	R
	до# (ре&)	S
	ре	T
	ре# (ми&)	U
	ми	V
	фа	W
фа# (соль&)	X	
соль	Y	
	пауза	Z

(В таблице знак «#» означает «диез», а «&» — «бемолю»).

Текст подпрограммы в машинных кодах:

НАЧ. АДРЕС : 036000

ДЛИНА : 000520

010346	006103	103020	005067	000340	010046	010146	010246
010446	010546	004767	000022	004767	000202	012605	012604
012602	012601	012600	012603	000207	012502	011501	060102
012704	036520	112100	120027	000056	001452	120027	000060
002444	120027	000065	003005	162700	000060	010067	000232
000434	120027	000100	003431	120027	000133	002026	012705
036354	112503	120300	001402	005725	000773	020427	037776
002013	004767	000306	005767	000160	001404	016700	000152
006303	077002	010324	000401	104420	020102	002007	000722
014400	010003	006203	060300	010024	000767	000207	012705
036520	020504	002041	012501	012502	005701	001414	010103
052737	000100	177716	077301	010103	042737	000100	177716
077301	077213	000415	012701	000100	010103	000240	000240
000240	077301	010103	000240	000240	000240	077301	077213
012703	001000	077301	000735	000207	000003	177501	041007
004360	161503	042010	004726	145105	043011	005277	132107
044013	005652	120111	045014	006627	107513	046016	007207
077515	047017	010170	070517	050021	011153	062521	051023
012537	055123	052026	013525	050125	053031	015113	043527
054034	017103	037531	055040	012000	000000	112503	042703
177400	010324	112503	042703	177400	000207	000000	000000

73

КОНТРОЛЬНАЯ СУММА : 053426₈

длительность влияет на все последующие ноты, пока не встретится другая цифра. Ноты кодируются заглавными латинскими буквами (каждой букве от А до Y соответствует одна из нот, как это показано в таблице), а пауза кодируется латинской буквой Z. Точка, стоящая после обозначения ноты, увеличивает ее длительность в полтора раза.

Некоторые подробности об алгоритме работы программы. Для обеспечения «непрерывности» исполнения мелодии в ней применен принцип компиляции нотного текста. Программа, обрабатывая заданный нотный текст, создает в оставшемся между ней и экраном объеме ОЗУ (адреса &036520 — &037776), используемом в качестве буфера, исполняемую последовательность задержек и длительностей, а затем обрабатывает

созданную последовательность, «не отвлекаясь ни на что другое». Соответственно, никакую другую USSR-подпрограмму или данные в указанную область ОЗУ заносить нельзя. Если же отведенного под буфер объема ОЗУ для генерации мелодии недостаточно, будет выдано сообщение об ошибке 16. В этом случае кодирующую ноты текстовую строку нужно разделить на две или три меньших.

Литература:

Кетков Ю. Л. «Диалог на языке Бейсик для мини- и микро-ЭВМ», «Наука», 1988, стр. 340.

Г. Бленд «Основы программирования на языке Бейсик в стандарте MSX», Москва, «Финансы и статистика», 1989.

Заключение рецензента

К большому сожалению пользователей БК все, что Усенков Д. Ю. высказал о версии Бейсика БК-0010.01, верно. Но и это еще не все! Любой из достаточно опытных пользователей может еще много чего добавить. Например, о работе со строками, массивами, графикой.

Усенков Д. Ю. частично устранил одну

из несправедливостей в отличии Бейсика БК от Бейсика MSX, предложив программу в кодах, музифицирующую БК-0010.01. Программа вполне работоспособна. Правда, кодировка нот и длительностей не совсем удобна. Лучше было бы придерживаться того, как это реализовано в Бейсике MSX.

Жариков Л. Н.

На Бейсике с помощью операторов PLAY и SOUND легко составить простые мелодии. Однако все они одноголосные, т. е. в конкретный момент времени звучит лишь одна нота. Большинство же музыкальных произведений 4—5-голосные. Как составить программу, исполняющую одновременно до 5 нот с различной длительностью? Можно ли это выполнить на Бейсике?

Читатель *М. Степницкий* (Г. Саранск)

74 Г. ПИМЕНОВ, Д. ПИМЕНОВ
(г. Мурманск)

Четырехголосый музыкальный редактор на Бейсике для БК-0010.01 (БК-11М)

В отличие от многих зарубежных бытовых компьютеров БК имеет существенный недостаток: у него отсутствует музыкальный процессор. Его отсутствие в некоторой степени возмещалось программированием одноголосных мелодий через контроллер магнитофона (регистр 177716). Появились музыкальный Бейсик, Фокал, «Меломан» и множество других программ для создания и исполнения компьютерной музыки. Среди них выделяется программа «Клавесин» В. Савина — трехголосный музыкальный редактор, но и он не смог удовлетворить взыскательного меломана. Сложную мелодию без искажений переложить не удается да и три голоса явно недостаточно.

Нами был разработан четырехголосый редактор на Бейсике. Имитацию музыкального процессора осуществляет программа в машинных кодах. Особенность этой программы в том, что управление регистром 177716 осуществляется встроенным таймером (адреса 177706, 177710, 177712). Перечислим некоторые достоинства редактора:

возможность задания любых длительностей, встретившихся в нотном тексте (триоли, квинтоли и т. д.);

выбор начального темпа в широких пределах;

изменение темпа в соответствии с указаниями в нотном тексте;

accelerando и *ritenuto* в точном соответствии с характером мелодии, что позволяет преодолеть основной недостаток известных музыкальных редакторов — монотонность звучания;

задание штрихов исполнения (легато, стаккато, нон-легато) в точном соответствии с характером произведения;

повторение любой части в любом месте (аналог реприз, но с большими возможностями);

возможность выделения громкости звучания одного из голосов;

настройка по камертону;

изменение тональности в широких пределах;

доступность изменения всех режимов для пользователя;

эффективный код после компиляции набранного текста (6—10 байт на один аккорд).

Для программирования мелодий, исполняемых «музыкальным процессором», выбран самый простой путь — запись нотного текста в числовом виде. Все оттенки

исполнения также кодируются числами.
1—52 — высота ноты. Число 1 соответствует ноте ми большой октавы, 2 — фа большой октавы, 3 — фа... и т. д. до соль третьей октавы;

53—60 — зарезервированы для возможного расширения диапазона;

61 — конец мелодии (с последующим нулем);

62—71 — длительность от 1/2 до 1/64;

72—80 — зарезервированы для задания недвухдолжных размеров (1/6, 1/10, 1/12 и т. д.). Пример приведен в программе:

81—83 — *ritenuto* разной степени;

85—87 — *accelerando*;

84 — отмена;

88 — стаккато;

89 — *non-legato*;

90 — *legato*;

91—99 — темп;

100—199 — выделение нотного блока повтора;

200—299 — вставка выделенного блока (см. нотный пример).

Текст набирается, начиная со строки 1000 через DATA. Все оттенки исполнения выставляются до аккорда и действуют до

следующего переопределения или отмены. Текст заканчивается числами 61 и 0 — указание компилятору и процессору закончить работу. Если нот в аккорде меньше четырех, он должен заканчиваться нулем. Пауза обозначается таким же образом, две одинаковые ноты звучат в унисон с увеличенной громкостью. Ноты в аккорде должны располагаться, начиная с самого низкого тона. Два одинаковых аккорда в режиме *legato* звучат как один с длительностью, равной сумме длительностей двух аккордов. Внимательный разбор приведенного примера поможет понять тонкости переложения. Выход из программы во время прослушивания через клавишу «@». Если темп и тональность вас не устраивают, их можно изменить по вашему вкусу (строки 80 и 100). На БК-11М из-за повышенной частоты темп и тональность будут выше (при одних и тех же исходных данных). По этой же причине и качество звучания на БК-11М лучше, чем на БК-0010.01.

Всех интересующихся компьютерной музыкой приглашаем переписываться: 183008, г. Мурманск, Молодежный проезд, 6, кв. 33, Пименовым.

75

```
5 REM **Четырехголосный музыкальный редактор**
6 REM **      БК-0010-01      **
7 REM **      " Л Е Л Ь - 4.03"      **
8 REM ** Авторы: Пименов Г. Н., Пименов Д. Г. **
9 REM **      17.01.1992 г. Мурманск      **
70 DIM A%(80%), E%(20%)
80 A%(62%) = 11000% 'начальный темп (максим. 37700)
90 K! = 1.059463! 'коэф. темпированного строя
100 F! = 256% 'тональность
110 FOR I% = 1% TO 52% 'массив нот (от МИ большой
120 A%(I%) = -(F! + .2!) 'до СОЛЬ третьей октавы
130 F! = F! / K! 'через полтона)
140 NEXT I%
150 FOR I% = 63% TO 71% STEP 2% 'массив стандартных
160 A%(I%) = A%(I% - 1%) * .75! 'длительностей
170 A%(I% + 1%) = A%(I% - 1%) / 2% '1/2, 1/4, 1/8, 1/8,
180 NEXT I% '1/16, 1/16, 1/32, 1/32, 1/64.
190 A%(74%) = A%(62%) / 3% 'длит. 1/6 (триоль четвертными)
200 FOR I% = 37400 TO 37742 STEP 2%
210 READ M% 'считывание программы в кодах
220 POKE I%, M% 'в память (строки 232 - 264)
230 NEXT I% '"музыкальный процессор"
232 DATA 3012700, 30100, 3010610, 3010037, 30177660, 305037, 30177706
234 DATA 3010706, 3062706, 30324, 3012737, 3020, 30177712
236 DATA 3020037, 30177662, 301004, 3011006, 305037, 30177660, 30207
238 DATA 3012767, 30177770, 30210, 3012767, 30177766, 30162, 3012767
240 DATA 3020115, 30142, 3012767, 3020215, 30154, 3012767, 3020315, 30166
242 DATA 3012767, 3020415, 30200, 3012601, 301004, 3012767, 30437, 30106
244 DATA 30432, 3012602, 301007, 3012767, 30427, 30112, 3012767, 30177772, 3076
```

```

246 DATA &0421, &012603, &01012, &012767, &0417, &0110, &012767, &0177770
248 DATA &054, &012767, &0177772, &066, &0405, &012604, &01003, &012767
250 DATA &0407, &0100, &012605, &01710, &06105, &05405, &010537, &0102
252 DATA &012705, &0177710, &020115, &0101406, &040037, &0177716
254 DATA &066601, &0177766, &050037, &0177716, &020215, &0101406
256 DATA &040037, &0177716, &066602, &0177770, &050037, &0177716
258 DATA &020315, &0101406, &040037, &0177716, &066603, &0177772
260 DATA &050037, &0177716, &020415, &0101746, &040037, &0177716
262 DATA &066604, &0177774, &050037, &0177716, &023715, &0102
264 DATA &0101735, &0630
270 S%=&037742
280 S%=S%+2% 'начальный адрес волнового блока
290 READ M%
300 IF M%<61% THEN 380 'считывание нот
310 IF M%>60% AND M%<81% THEN L%=A%(M%) 'длительность
320 IF M%>80% AND M%<88% THEN C%=1%+(M%-84%)/100%'ritenuto и acceler.
330 IF M%>90% AND M%<100 THEN T%=1%+(M%-91%)/3%'темп
340 IF M%=88% THEN R%=2%'стаккато
350 IF M%=89% THEN R%=1.1%'нон-легато
360 IF M%=90% THEN R%=1%'легато
362 IF M%>99% AND M%<200% THEN E%(M%-100%)-S%'метки для повтора
364 IF M%>199% THEN 522'повтор по меткам
370 GOTO 290
380 POKE S%, A%(M%)
390 IF M%=0% THEN 420
400 M%=M%+1%
410 IF M%>4% THEN 280
420 M%=0%
430 S%=S%+2%
440 POKE S%, L%/R!/T!
450 IF L%=0% THEN 540'запуск программы на исполнение
460 IF R!>1% THEN 490'паузы для стаккато и нон-легато
470 T%=T!#C!
480 GOTO 280
490 POKE S%+2%, 0%
500 POKE S%+4%, L%/T!-L%/T!/R!
510 S%=S%+4%
520 GOTO 470
522 FOR I%=E%(M%-200%) TO E%(M%-199%)-2% STEP 2%
524 POKE S%, PEEK(I%)
526 S%=S%+2%
528 NEXT I%
530 GOTO 290
540 DEF USR=&037400
550 I%=USR(0%)
560 IF PEEK(&0177662)=64 THEN END ELSE 550'выход по клавише "e"
570 DATA 66, 84, 90, 91'начальная установка
1000 '***** Л А Н Б А Д А *****
1001 DATA 11, , 35, 0, 89, 14, 18, 23, 35, 14, 18, 23, 35, 14, 18, 23, 33
1002 DATA 103, 90, 11, , 26, 0, 89, 14, 18, 23, 26, 14, 18, 23, 26, 14, 18
1003 DATA 23, 30, 90, 11, , 37, 0, 89, 14, 18, 23, 37, 14, 18, 23, 35, 14, 18, 23
1004 DATA 33, 90, 11, , 26, 0, 89, 14, 18, 23, 26, 14, 18, 23, 30, 14, 18, 23, 33
1005 DATA 90, 4, , 32, , 68, 89, 16, 20, 23, 32, 16, 20, 23, 30, 66, 16, 20, 23, 28
1006 DATA 16, 20, 23, 28, 4, , 28, , 16, 20, 23, 28, 16, 20, 23, 30, 16, 20, 23, 28

```

1007 DATA 104, 90, 6, , 30, , 13, 18, 21, 30, 6, , 30, 0, 13, 18, 21, 30, 6, , 0, 89
 1008 DATA 13, 18, 21, 0, 13, 18, 21, 0, 68, 90, 13, 18, 21, 0, 69, 13, 18, 21, 0
 1009 DATA 70, 37, , 0, 66, 89, 11, , 35, , 101, 14, 18, 23, 35, 102, 201, 14, 18
 1010 DATA 23, 33, 203, 90, 68, 6, , 33, , 6, , 32, , 66, 13, 18, 21, 30, 6, 30, 0, 13
 1011 DATA 18, 21, 30, 89, 6, , 0, 101, 13, 18, 21, 0, 102, 201, 201, 105, 90
 1012 DATA 6, , 37, , 13, 18, 21, 37, 89, 13, 18, 21, 37, 13, 18, 21, 35, 6, , 33, , 13
 1013 DATA 18, 21, 32, 90, 13, 18, 21, 30, 89, 13, 18, 21, 30, 2, , 30, , 14, 18, 21
 1014 DATA 33, 14, 18, 21, 32, 14, 18, 21, 30, 4, , 28, , 16, 20, 23, 30, 16, 20, 23
 1015 DATA 25, 16, 20, 23, , 106, 90, 9, , 25, , 89, 68, 13, 16, 21, 25, 13, 16, 21, 23
 1016 DATA 66, 13, 16, 21, 25, 68, 16, 21, 33, 37, 35, 38, 0, 9, , 37, 40, 37, 40, 0
 1017 DATA 13, 16, 21, 0, 35, 38, 0, 13, 16, 21, 0, 33, 37, 0, 66, 16, 21, 37, 40, 205
 1018 DATA 61, 0

"Музыкальный процессор под микроскопом МИКРО10"

: Четырехголосие на БК 0010-01 с использованием
 ; встроенного таймера.

; *****
 ; * Мурманск 17-01-92 Пименов Г. Н. "ЛЕЛЬ 4.03" *
 ; *****

77

MOV #100, R0 ; занести указатель стека
 MOV SP, (R0) ; в ячейку #100
 MOV R0, @#177660 ; запретить прерывание от клавиатуры
 CLR @#177706 ;
 MOV PC, SP ; адрес начала
 ADD #324, SP ; мелодии
 NO: MOV #20, @#177712 ; запуск таймера
 CMP R0, @#177662 ; выход по клавише "e"
 BNE E1 ;
 E2: MOV (R0), SP ; восстан. стека
 CLR @#177660 ; разрешение прерывания
 RTS PC ;
 E1: MOV #177770, R1+2
 MOV #177766, R0+2 ; начальная модификация
 MOV #20115, A3 ; ячеек
 MOV #20215, A4 ;
 MOV #20315, A5
 MOV #20415, A6
 MOV (SP)+, R1 ; первую ноту в R1
 BNE M1 ;
 MOV #437, A3 ; иначе модификация A3 (пауза)
 BR M4 ; и переход на M4
 M1: MOV (SP)+, R2 ; вторую ноту в R2
 BNE M2

MOV #427, A4	; иначе модификация A4,
MOV #177772, NO+2	; NO+2
BR M4	;
M2: MOV (SP)+, R3	; третью ноту в R3
BNE M3	;
MOV #417, A5	; модификация ячеек
MOV #177770, NO+2	
MOV #177772, N1+2	
BR M4	
M3: MOV (SP)+, R4	; четвертую ноту в R4
BNE M4	;
MOV #407, A6	
M4: MOV (SP)+, R5	; длительность в R5
BEQ E2	;
M5: ROL R5	; увеличить длительность
NEG R5	; в 2 раза и изменить знак
MOV R5, @#102	
MOV #177710, R5	; счетчик таймера в R5
A3: CMP R1, (R5)	; сравнить 1-ю ноту со
BLOS A4	; счетчиком. Если содержимое
BIC R0, @#177716	; счетчика ниже, очистить 6-й
NO: ADD -12(SP), R1	; разряд, "восстановить" ноту,
BIS R0, @#177716	; установить 6 разряд
A4: CMP R2, (R5)	; сравнить 2-ю ноту
BLOS A5	;
BIC R0, @#177716	;
M1: ADD -10(SP), R2	;
BIS R0, @#177716	;
A5: CMP R3, (R5)	; сравнить 3-ю ноту
BLOS A6	;
BIC R0, @#177716	;
ADD -6(SP), R3	;
BIS R0, @#177716	;
A6: CMP R4, (R5)	; сравнить 4-ю ноту
BLOS A3	;
BIC R0, @#177716	;
ADD -4(SP), R4	;
BIS R0, @#177716	;
A7: CMP @#102, (R5)	; проверка окончания
BLOS A3	; длительности
BR NO	; на считывание след. аккорда

; Начало четырехголосной мелодии

Программирование мелодий на БК-0011М

В Бейсике БК-0011 был очень хороший оператор выдачи звука ВЕЕР, с помощью которого школьники могли без труда создавать звуковое сопровождение для своих программ. БК-0011М снизошел до БК-0010 и, кроме короткого звукового сигнала, ничего выдавать не хочет. Остается два выхода — музыкальный редактор или программа в кодах. О первом говорить нечего — он еще только создается. Машинные коды заводчане постарались изменить, не сказав об этом. Однако эта информация полезна.

В машинных командах есть команда EMT-101, не описанная в руководствах, которая выдает звуковой сигнал в соответствии с содержимым регистров R0 и R1, где R0 — длительность, а R1 — высота звука (для БК-0010, 0011 это EMT-124). Значения операндов оператора ВЕЕР у БК-0011 совпадают со значениями операндов команды EMT-101.

Коды высоты звучания для нот первой октавы (восьмеричные/десятичные):

до	— 644/420
до диез	— 614/396
ре	— 566/374
фа диез	— 450/296
соль	— 430/280
соль диез	— 410/264
ре диез	— 540/352
ми	— 515/333
фа	— 472/314
ля	— 372/250
ля диез	— 353/235
си	— 336/225

Для получения кодов нот других октав необходимо учитывать, что они отличаются от соответствующих кодов нот соседних октав в 2,4 и т. д. раз.

Пример:

- ля первой октавы — 250
- ля второй октавы — $250 \times 2 = 500$
- ля малой октавы — $250/2 = 125$.

Код длительности звучания определяется исходя из того, что произведение кода длительности на код высоты для нот одной длительности должно быть одинаково, в зависимости от заданного темпа.

Пример:

- 1/8 ля первой октавы — высота — 250, длительность — 160

$$250 \times 160 = 40000$$

- 1/8 си первой октавы — высота — 222, длительность —

$$40000/222 = 180$$

1/8 до первой октавы — высота — 420, длительность —

$$40000/420 = 95.$$

Подпрограмма, воспроизводящая мелодию, состоит из трех частей:

- 1) подпрограмма воспроизведения;
- 2) коды длительностей нот;
- 3) коды высот нот.

Первая часть загружается один раз и вызывается по мере необходимости. Вторую и третью части, а также содержимое ячейки &O1012 изменяют в зависимости от мелодии.

Подпрограмму воспроизведения легче записать в мониторе, выйдя в него из Бейсика по команде MON, заполнить ячейки, вернуться в Бейсик, нажав клавишу P (лат.), определить ее как функцию на машинном языке с помощью оператора DEF USR=
=&O1000 и вызывать по мере необходимости с помощью A=USR(A).

Подпрограмма воспроизведения в машинных кодах:

- 1000/12703 — запись ячейки, с которой начинаются коды длительностей (1100)
- 1002/1100
- 1004/12704 — запись ячейки, с которой начинаются коды высот (1200)
- 1006/1200
- 1010/12702 — запись количества нот (12) в R2
- 1012/12
- 1014/12300 — запись кода длительности в R0
- 1016/12401 — запись кода высоты в R1
- 1020/104101 — команда воспроизведения звука
- 1022/77204 — ветвление на ячейке 1014, если $R2 > < 0$
- 1024/2 — возврат из подпрограммы.

Подпрограмма воспроизведения на Бейсике:

```
10 DATA &O12700,&O1100,&O12704,&O1200,&O12702,&O12,
&O12300,&O12401
20 DATA &O104101, &O77204,&O2
30 FOR I=&O1000 TO &O1024 IFT 2K
40 READ TX
50 FORK IX, TX
60 NEXT
70 DEF USR=&O1000
80
```

90 Коды длительностей для "Подмосковные вечера"
 100
 110 DATA 160,190,241,190,448,190,180,482,428,640
 120 FOR IX=&O1100 TO &O1122 ST ZX
 130 HEAD TX
 140 FORC IX, TX
 150 NEXT
 160
 170 Коды высот
 180
 190 DATA 250,210,166,210,187,210,222,166,187,250
 200 FOR IX=&O1200 TO &O1222 ST ZX
 210 HEAD TX
 220 FORC IX, TX
 230 NEXT
 240 Востроизведение мелодии
 250 A=UER(A)

Команда EMT 31 отключает звук при нажатии на клавишу, если в регистре R0 находится 100 (восьмеричное). Включение звука команды EMT 31 осуществляет при R0 отличном от 100, т. е. используется шестой разряд.

Программа отключения звука в кодах:
 1000/12700 — запись в R0 кода 100
 1002/100
 1004/104031 — отключение звука
 1006/2 возврат из подпрограммы на Бейсике:

10 DATA &O12700,&O100,&O104031,&O2
 20 FOR IX=&O1000 TO &O1006 ST ZX
 30 HEAD TX
 40 FORC IX, TX
 50 NEXT
 60 DEF UER=&O1000
 70 A=UER(A)

Отключение звука при помощи команды EMT 31.

80

В. МАСЛОВ
 (г. Калуга)

Музыка на Форте

Язык Форт удобное средство для создания музыкальных программ. Для примера рассмотрим реализацию простейшего языка для записи мелодий. Эта же программа, написанная на Паскале или Си, займет у вас около 50 строк.

Единственный оператор, который генерирует звук, это SOUND. Но в нем не совсем удобные параметры.

Первый параметр слова SOUND равен количеству полуволи, а второй — длине полуволи. Это не удобно при вводе данных, поэтому введем слово MODIF, которое будет пересчитывать длительности нот в зависимости от длины волны (частоты).

: TEMPO 32000; (Константа темпа)
 : MODIF DUP ROT*TEMPO10ROT*/,,;

Теперь можно вводить:

OK 16 100 MODIF SOUND

Здесь 16 — длительность, 100 — частота.

Длительности нот будем обозначать целыми числами:

- 1 — целая
- 2 — половинная
- 4 — восьмая
- 16 — ... и т. д.

Данная запись удобна и для дробных длительностей. Так триоль четвертой ноты будет равна:

$1/4/3=1/12$, т. е. в нашей записи 12.

С помощью слова TONE определяем названия нот и их длительности. Для этого перед TONE ставится частота, а после TONE обозначение ноты.

Например:

100 TONE C

: TONE CREATE, DOEC> @MODIF;

После слова DOES> определяется действие слова, регламентированного текущим определением, т. е. словом TONE. Иначе говоря, мы определяем действие введенных нот. Они при исполнении будут выполнять команду MODIF, т. е. пересчитывать частоту и длительность под SOUND.

PLAY — последнее слово. Оно получает в стеке адрес массива нот и проигрывает их с помощью SOUND.

Пример:

OK CREATE M 8 C 8D 8 E 8 G0,

OK M PLAY

Попробуйте усовершенствовать язык, введя дополнительные удобства при определении мелодии. Например, следующим образом:

OK MUSIC...ENDM

Ниже приводится полный текст музыкального языка.

Экран 1

```
0 FORGET K
1 : K ;
2 : TEMPO 32000 ;
3 : MODIF DUP ROT * TEMPO 10 ROT */ , , ;
4 : PLAY BEGIN DUP @ DUP
5 WHILE SWAP 2+ DUP @ ROT SWAP SOUND 2+ REPEAT 2DROP ;
6 : TONE CREATE , DOES> @ MODIF ;
7 : .. HERE 2- DUP 2- @ , @ , ;
8
9 250 TONE C 223 TONE D 198 TONE E 187 TONE F 167 TONE G
10 149 TONE A 140 TONE H 132 TONE B 125 TONE 2C
11
12
13
14
15
```

81

ОБМЕН ОПЫТОМ

Р. РАХМАНКУЛОВ

(г. Ташкент)

Звуковые эффекты на БК-0010

Для создания звуковых эффектов в программах на Ассемблере можно воспользоваться следующей программой. Текст приводится на Турбо-ассемблере.

```
SOUND: MOV @(R5)+,R0
        MOV @(R5)+,R1
        MOV @(R5)+,R2
        MOV #177716,R3
1:      MOV #300,(R3)
        MOV R0,R4
2:      SOB R4,2
        MOV #200,(R3)
        MOV R1,R4
3:      SOB R4,3
        SUB D1,R1
        SUB D2,R1
        SOB R2,1
        RTS R5

T1:     .WORD 1000
T2:     .WORD 1000
T3:     .WORD 700
D1:     .WORD 1
D2:     .WORD 1
```

Меняя параметры T1, T2, T3, D1, D2, а также константы, засылаемые в регистр 177716, можно подобрать желаемые звуковые эффекты. Вызвать подпрограмму можно следующим образом:

```
JSR R5,SOUND
.WORD T1
.WORD T2
.WORD T3
```

Программу можно упростить, если принять T1-T2 и D1-D2. Возможности уменьшатся, но останутся достаточно широкими. В этом случае программу можно переписать следующим образом:

```
SOUND: MOV @(R5)+,R0
        MOV @(R5)+,R2
        MOV @(R5)+,R1
        MOV #177716,R3
1:      MOV #300,(R3)
        MOV R0,R4
2:      SOB R4,2
        MOV #200,(R3)
        MOV R0,R4
```

3: SOB R4,3
 SUB R1,R0
 SOB R2,1
 RTS R5

Обращаться к ней

JSR R5,SOUND
 .WORD T1
 .WORD T3
 .WORD D1

T1 — условная начальная высота,
 T3 — условная длительность,
 D1 — условная скорость изменения тона:
 если D1 больше 0, то тон звука
 повышается;
 если D1 меньше 0, то тон звука
 понижается;
 если D1 равна 0, то тон звука не
 изменяется.

82 После небольшой переделки эту программу можно использовать в программах на Бейсике. Но в программах на Бейсике и подпрограмме в кодах передается только один аргумент. В этом случае можно поступить, например, так: подобрать параметры для 5—10 или более понравившихся вам звуков и в программе обращаться по номеру. Соответствующая программа на Бейсике выглядит так:

```
10 DATA &012703,&037570,&011501,&020127,0,&01403,&062703,
&06,&077103,&04767,2,&0207,&012300,&012302,&012301,
&012703,&0177716,&012713,&0300,&010004,&077401,
&012713,&0200,&010004,&077401,&0160100,&077212,&0207

20 DATA &0400,&050,0,&010,&05000,0,&050,&01000,0,&02000,&02000,
1,1,&01000,&0177777,&0100,&01000,0,&01000,&0400,0
&03000,&01000,2,&01000,&0200,4,1,&0100,&0177777,
1,&0300,&0177770,&0300,&01000,0,&070,&01000,0
&0150,&01000,0

30 DATA &0200,&01000,0,&0200,&0500,0,&0400,&0300,0,&0500,
&0300,0,&0600,&0300,0,&0700,&0300,0,&01000,&0200,0

40 FOR I%=1 TO 91

50 READ A%

60 POKE &037476+I%*2%,A%

70 NEXT I%

80 DEF USR=&037500

90 FOR I%=0 TO 20

100 A%=USR(I%)

110 NEXT I%

120 END
```

Оператор 10 — машинный код программы
 20—30 — параметры для подпрограммы по 3 — T1, T3, D1 для 1 % от 0 до 20
 40—70 — занесение программы и данных в память с адреса 37500
 90—110 — прослушивание звуков.

При желании, конечно, можно изменить параметры. Можно менять параметры непосредственно во время выполнения программы (оператором POKE). Можно подобрать параметры для того, чтобы сыграть мелодию. Нетрудно также переписать программу для задания параметров в виде: нач. частота, кон. частота, шаг. и т. д.

С. ИВАШИННИКОВ

(г. Арсеньев, Приморский край)

О создании музыкального оформления

Эта простенькая программа принесет приятные минуты и начинающим, и умудренным пользователям БК.

Заключение рецензента

Предлагаемая программа использует особенности работы трех регистров БК, позво-

1000А

012737, 000140, 177706, 012737, 000040, 177712, 012700,
000004, 012701, 002000, 005341, 003402, 077003, 000770.
013711, 177710, 052737, 000100, 177716, 042737, 000100,
177716, 000757.

После набора представленных 23 машинных слов музыкальной программы запустите ее с адреса 1000. Оценив эффект, попробуйте экспериментировать. В приведенном фрагменте ячейка 1016 имеет значение 4, что формирует 4 голоса (4 звуковых псевдоканала). Диапазон значений этой ячейки 1 — 340. Изменяя содержимое ячейки 1002, можно управлять звуковой октавой эффектов. Наконец, ячейка 1010 может принимать значения 20, 40, 100, 200, 60, 140, 300, 160, 340, 360. Фактически эта программа позволяет исследовать работу трех регистров незадокументированного таймера БК. Регистр 177706 определяет диапазон счетчика 177710, а регистр 177712 управляет скоростью работы таймера.

Известно, что БК имеет лишь один голос, что явно обедняет его возможности по сравнению с музыкальными возможностями многоканальных компьютеров. Быть может, эта программа подвигнет заинтересованных на создание многоголосых мелодий для БК. При этом не потребуются никаких аппаратных доработок компьютера.

Д. ПОПОВ
(Москва)

Говорящий «Апогей»

Мне захотелось услышать голос моего персонального компьютера «Апогей БК-01» (для справки: к семейству компьютеров БК не относится, он ближе к «Микроше» и «Синклеру»). Для этого я написал небольшую программу.

ляющих организовать таймер. Работа с этими регистрами, имеющими восьмеричные адреса 177710, 177706, 177712, достаточно подробно описана в журнале «Информатика и образование», № 1, 1990.

Музыкальные возможности программы автор, конечно, сильно преувеличил, но с помощью этой программы можно создать несколько необычных звуковых эффектов. Например, эффект скачущего теннисного шарика достигается при содержимом ячеек программы 1002, 1010, 1016, равному 2000, 160,5 соответственно. Программа загружается с адреса 1000. Для создания эффекта «заточки ножа» необходимо в ячейки по указанным адресам поместить 600, 60, 4 соответственно. Упорным поиском можно подобрать и другие звуки. Созданные эффекты можно затем включить в прикладные или игровые программы.

Жариков Л. Н.

Она состоит из двух частей — записывающей и воспроизводящей, занимает в памяти 112 байт и позволяет записывать фонограммы длительностью порядка 75—80 с.

С помощью директивы монитора М или

какого-либо отладчика введите в память с адреса 0000H коды программы. Проверьте частичные и полную контрольные суммы, исправьте ошибки, которые, возможно, возникли при вводе.

Подключите к вашему магнитофону микрофон, нажмите клавиши «Пауза», «Пуск» и «Запись». Запустите программу командой G и говорите в микрофон до появления на экране стандартного приглашения монитора.

Для воспроизведения звука запустите программу командой G3, не выключая магнитофон.

Длительность фонограммы можно увеличить за счет ухудшения качества звука, увеличив содержимое ячейки 005EH.

Следует помнить, что фонограмма занимает всю оперативную память.

Листинг программы воспроизведения человеческого голоса:

```

0000:C3 2B 00 31 6F 00 21 70 00 11 FF E0 46 0E 0B 7B . 6FE3
0010:17 47 3E 01 17 32 02 ED CD 5C 00 E5 E1 0D C2 0F . 9BA2
0020:00 23 CD 56 00 C2 0C 00 C3 75 FB 31 6F 00 21 70 . 0A75
0030:00 11 FF E0 01 0B 00 3A 02 ED E6 10 17 17 17 17 . 6174
0040:7B 17 47 CD 5C 00 0D C2 37 00 70 23 CD 56 00 C2 . BF7D
0050:34 00 C3 75 FB 00 7C BA CD 7D B8 C9 C5 06 0B 05 . 3533
0060:C2 5F 00 C1 C9 20 2B 43 29 20 50 56 41 5B 20 00 . E2DE

```

Полная контрольная сумма 21FCH.

* * *

В заключение немного о специфических особенностях использования Бейсика «Микрон» на компьютере «Апогей».

В. ЛУКИЧЕВ

Звуковое оформление программ

УКНЦ имеет схему генерации звукового сигнала, и вы можете программным путем воспроизводить звуки нужной вам частоты. Для этого необходимо использовать системный регистр управления 177716 (магистраль ПП). В этом регистре биты с 8

Оператор ВЕЕР дает очень тихий звук. Для получения более громкого звука и сложных звуковых эффектов целесообразнее обращаться к таймеру КР580ВИ53. По адресам ЕС00Н, ЕС01Н, ЕС02Н находятся ячейки счетчиков таймера. Загружая в них числа, можно получать звуковые сигналы, непрерывно воспроизводимые параллельно работе центрального процессора. По адресу ЕС03Н находится управляющий регистр таймера.

Сформировать звуковой сигнал можно с помощью последовательности команд:

```

POKE—5120+CH, F:POKE—5120+CH, F,

```

где CH (0..2) — номер звукового канала (счетчика таймера), F (0..255) — число, обратно пропорциональное частоте звука.

«Выключается» звук (по всем каналам) другой последовательностью команд:

```

POKE—5117,62:POKE—5117,126:POKE—
—5117,190.

```

Аналогично можно управлять звуком и в программах на языке Ассемблер.

Прием, используемый в ПК «Микроша» (обращение к контроллеру клавиатуры для управления счетчиком 2 таймера), на ПК «Апогей» не проходит.

по 12 отвечают за прохождение соответствующих звуковых частот, а бит 7 — за включение/выключение сигнала. Сбрасывая и устанавливая бит 7, можно программно промодулировать звук необходимой частоты. Тактовая частота программируемого

таймера также влияет на частоту извлекаемого звука, при выключенном таймере звука не будет. Программа заставляет УКНЦ воспроизвести несложный звуковой ряд:

```

...
MOV PC,R0 ; предшествующая часть программы
ADD #STRING-. ,R0
1$: TSTB (R0) ; получить в R0 адрес строки с нотами
BEQ 33$
JSR PC,WAIT5 ; еще не конец ?
; если да, то переход на метку OUT
JSR PC,SOUND ; найти ноту в таблице и получить в
INC R0 ; R2 задержку, соответствующую этой ноте
BR 1$ ; проиграть эту ноту
WAIT5: ; увеличить указатель
; возврат на начало цикла
MOV PC,R1 ; подпрограмма получения задержки
ADD #TBL-. ,R1 ; получить в R1 адрес таблицы нот
10$: CMPB (R1)+,(R0)
BEQ 11$ ; сравнить
BR 10$ ; она - переход на метку 11$
11$: MOVB (R1),R2 ; не она - возврат на метку 10$
BIC #177400,R2
RTS PC ; поместить задержку в R2
SOUND: ; очистить старший байт R2
MOV #100.,R1 ; возврат из подпрограммы
10$: BIS #200,@#177716 ; подпрограмма воспроизведения ноты
MOV R2,R3 ; продолжительность звучания ноты
SOB R3, ; включение звука
BIC #200,@#177716 ; задержку из R2 в R3
MOV R2,R3 ; пустой цикл
SOB R3, ; выключение звука
SOB R1,10$ ; задержку из R2 в R3
RTS PC ; пустой цикл
OUT: ; если R1 не равен 0, то на метку 10$
; возврат из подпрограммы
STRING: .ASCIZ "CDEFGAH12345" ; продолжение программы
.EVEN
TBL: .BYTE 'C,177
.BYTE 'D,161
.BYTE 'E,145
.BYTE 'F,137
.BYTE 'G,125
.BYTE 'A,113
.BYTE 'H,103
.BYTE '1,77
.BYTE '2,70
.BYTE '3,62
.BYTE '4,57
.BYTE '5,52
.BYTE ',47
0

```

Ф. НОВИКОВ
(г. Ярославль)

Подпрограмма в ПЗУ

Немногие пользователи БК знают, что в ПЗУ монитора защита подпрограмма звуковых эффектов. При ее работе после нажатия клавиши раздается булькающий звук. Занеся во 2-й и 3-й регистры и обратившись к ней по адресу 102062, можно получить интересные эффекты.

Например

[1] : A4L

```
MOV #10,R2
MOV #200,R3
JEB FC,@#102062
```

86

В. НИКОЛАЕВ
(г. Климовск)

Рев БК

По мнению некоторых пользователей, существенным недостатком БК-0010.01 является малая мощность встроенного звукогенератора. Действительно, разве можно в игровой программе считать выстрелами едва слышные потрескивания, а взрывами — жалкие попискивания, доносящиеся из компьютера?

Между тем этот недостаток легко устраним, так как при выполнении программы сигнал звуковой частоты поступает не только на звукогенератор, но и на разъем МГ (ножка 5) компьютера. Следовательно, для усиления звука нужно лишь усилить электронный сигнал с этой ножки каким-либо внешним устройством и акустически его излучить.

В простейшем случае таким устройством может быть магнитофон, используемый для загрузки программ. Если он позволяет прослушивать (с усилением) записываемую фонограмму, включите его (кабелем, входящим в комплект БК) на запись от разъема МГ. Естественно, кассету с программами предварительно надо заменить какой-либо ненужной кассетой. Используйте режим «стоп-ленты» (если он предусмотрен в магнитофоне), что устранил износ ленты и головок.

Если этот вариант усиления звука для вас неосуществим, вспомните, что режим «усиление внешнего сигнала» имеется во многих других устройствах бытовой электро-

Содержимое R2 должно быть всегда меньше содержимого R3!

Подобную программу можно написать на Бейсике:

```
10 DATA #012702, #010, #012703, #0200, #04737, #0102062, #0207
20 FOR I=#037400 TO #037414 STEP 2
30 READ A
40 FORK I, A
50 NEXT I
60 DEF USB=#037400
70 A=USR(A)
```

ники. Например, как правило, он имеется в проигрывателях.

Наконец, можно смастерить самодельный усилитель с динамиком [1], используя для питания +5В с блока питания компьютера, или завести электронный сигнал в тракт звуковой частоты телевизора, который чаще всего используется в качестве монитора. Однако эти варианты сложны и автором не опробовались.

Итак, попробуйте усилить звуковые эффекты выполняемых программ! Возможно, это увеличит их привлекательность для пользователя.

Литература

1. Янцев В. Минимум деталей и полчаса времени — и в ваших руках усилитель низкой частоты // Юный техник.— 1991.— № 9.— С. 16.

Д. УСЕНКОВ

От БК-0010 до БК-0011М

Первая информация о БК-0011 и о других перспективных моделях (например, БК-0100) появилась в журнале «Наука и жизнь» в 1987 году (№ 8, с. 33). Тогда о БК-0011 говорилось, как о разрабатываемой модели и приводились данные скорее желаемые, чем реальные. Большую ясность в этот вопрос внес журнал «Вычислительная техника и ее применение» № 7 за 1988 год. В опубликованной там статье приводилось более подробное описание, однако оно по сути касалось только технических данных компьютера. Позже в журналах приводились лишь отрывочные, часто противоречивые сведения о БК-0011.

В одном из журналов кто-то из авторов посетовал, что БК-0011, мол, никак не совместима с БК-0010. Однако на самом деле БК-0011 совместима с БК-0010 не только через Бейсик (что довольно легко сделать даже на ЭВМ с разными процессорами), но и по машинным командам. Форматы записи на магнитную ленту у БК-0011 и БК-0010 аналогичны. Это позволяет нормально загружать и запускать на выполнение практически любую программу, будь то в кодах, или на Бейсике, написанную для БК-0010. В последнее время в печати появились статьи с описанием средств, совмещающих БК-0010 с БК-0011. Учитывая это, можно не излагать здесь довольно объемистое описание БК-0011, а ограничиться описанием только различий между БК-0010 и БК-0011.

Технические характеристики

БК-0011 существует в пяти различных исполнениях с различными назначениями, комплектностью и стоимостью.

1. «Бытовой компьютер»: в комплект

входят информационно-вычислительный блок (клавиатура и процессорная электроника компьютера; для краткости будем далее называть этот блок просто «клавиатура»), блок питания, блок МСТД, отдельный блок контроллера принтера, кассета с программами на Бейсике (кстати, все с теми же, что и для БК-0010), соединительные шнуры и руководство для пользователя (описание системы, монитора и Бейсика).

2. «РМУ (рабочее место ученика)»: клавиатура, блок питания, черно-белый монитор, контроллер локальной сети (внешний блок), кабели и руководство пользователя.

3. Базовое РМП (рабочее место преподавателя): клавиатура, блок питания, блок МСТД, черно-белый монитор, принтер, сдвоенный дисковод, контроллер принтера, диски с ОС БК-0011, Бейсиком, Паскалем, Микроассемблером и чистые, кабели и полный комплект документации по всем перечисленным языкам, а также по системе, устройству и системному программированию для БК-0011. В этом же комплекте прилагаются все принадлежности, необходимые для создания на базе БК-0011 компьютерного класса.

4. «Дополнительное РМП»: клавиатура, блок питания, МСТД, контроллеры принтера и дисковода, кабели, кассета с программами, диски с ОС БК-0011 и чистые, набор документации.

5. «Профессиональная ЭВМ»: клавиатура, блок питания, МСТД, черно-белый монитор, принтер, сдвоенный дисковод, контроллеры принтера и дисковода, кассета с программами, диски с ОС БК-0011 и чистые, кабели и полный комплект документации.

ПРИМЕЧАНИЕ: в комплектах везде указан черно-белый монитор, но это не озна-

чает, что к БК-0011 нельзя подключить цветной монитор типа используемого для БК-0010 32ВТЦ201. Вообще же вся информация о подключении БК-0010 к телевизору верна и для БК-0011.

Краткие технические характеристики БК-0011:

тип и разрядность процессора	— К1801ВМ1, 16-разрядный
быстродействие, тыс. оп/с	— 500
объем ОЗУ, кБ	— 128
(в том числе экранного)	— 32 (2 страницы)
объем ПЗУ, кБ	— 40
(плюс внешнее МСТД)	— 8
емкость диска	— от 400 кБ до 800 кБ, в зависимости от дисковода
организация памяти	— страничная, по 16 кБ
строк текста на экране	— 25 (1 служебная)
точек в строке	— 512, 256 или 128
количество клавиш клавиатуры	— 74
скорость обмена по локальной сети	— 57600 Бод
скорость записи на магнитофон	— 1200 Бод
потребляемая мощность	— около 40 Вт
масса	— не более 5 кг

88

(данные о массе и потребляемой мощности приведены для базового комплекта в исполнении 1).

Внешний вид блоков клавиатуры, МСТД и питания не отличается от таковых для БК-0010. Расположение, количество и прорисовка клавиш клавиатуры полностью аналогичны БК-0010.01.

Монитор

Адресное пространство БК-0011 составляет, как и в БК-0010, 64 кБ (оно ограничено разрядностью адресной шины процессора). Однако, в отличие от БК-0010, здесь к одним и тем же областям адресов адресного пространства (оно называется виртуальным) могут быть подключены несколько различных «реальных» блоков ОЗУ, один из которых «явный», а остальные — «скрытые». По умолчанию процессор относит указанный адрес к ячейкам «явного» блока ОЗУ (эти блоки называются страницами памяти и могут включать в себя более мелкие «подблоки» — окна памяти), но возможен вызов подпрограмм и данных непосредственно со скрытых страниц, а также обмен данными между любыми страницами. Распределение адресного пространства БК-0011 показано на рис. 1.

На экран обычно отводится две страницы ОЗУ, но видна на экране монитора в текущий момент только одна из них. Возможен и такой вариант: пока одна из страниц демонстрируется на экране, во вторую, невидимую страницу, производится вывод новой информации.

В отличие от БК-0010, где при включении питания сразу же запускается один из языков программирования, в БК-0011 при включении начинает работать Монитор. Функции Монитора БК-0011 по сравнению с Монитором БК-0010 значительно расширены, и, по сути, включают в себя возмож-

ности режима «ТС». Из МСТД монитора пользователь может при необходимости запустить транслятор Бейсика (с адреса 100000) или тесты (с адреса 160000). Набор тестов также значительно расширен по сравнению с БК-0010.

Тест 1 — тест диспетчера страниц.

Тест 2 — тест ОЗУ (постраничный/общий, циклический/однократный).

Тест 3 — тест ПЗУ.

Тест 4 — тест клавиатуры.

Тест 5 — тест порта ввода-вывода.

Тест 6 — тест вывода символов на экран.

Тест 7 — тест магнитофона.

Тест 8 — тест палитры (вывод палитры по заданному номеру).

Тест 9 — тест печати на принтер.

Монитор БК-0011 выводит при работе символ — приглашение «@», а при возникновении ошибки (отсутствие регистра, неподключенная страница и т. д.) выводится

ВИРТУАЛЬНОЕ АДРЕСНОЕ ПРОСТРАНСТВО 64 кБ

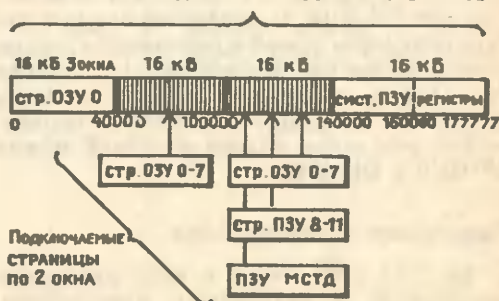


Рис. 1

признак ошибки «?». Монитор позволяет выполнить следующие операции:

- ⟨хххххх⟩/ — открыть ячейку с адресом хххххх (вывод на экран восьмеричного содержания ячейки)
- ⟨хххххх⟩\ — открыть байт
- R⟨х⟩/ — открыть регистр (⟨х⟩ — номер регистра 0—7 или «S» для чтения PSW)
- /и\ — открыть текущую ячейку (последнюю открытую)
- ⟨ВВОД⟩ — закрыть ячейку, байт или регистр. Если было введено новое числовое значение, оно переписывается в ячейку, байт или регистр
- ↓ — закрыть ячейку и открыть следующую по порядку
- ↑ — закрыть ячейку и открыть предыдущую
- — закрыть ячейку и открыть следующую, адрес которой вычисляется по формуле: ⟨Адрес⟩ = ⟨Текущий адрес⟩ + 2 + ⟨Содержимое прежней ячейки, возможно, новое⟩
- ← — то же, но адрес вычисляется по формуле: ⟨Адрес⟩ = ⟨Текущий адрес⟩ + ⟨Младший байт ячейки⟩ * 2 + 2
- @ — то же, но адрес равен содержимому текущей ячейки
- N; MC — отобразить страницу N с адреса, указанного в аргументе M
- ⟨хххххх⟩ L — загрузить файл с магнитофона по адресу хххххх или, если он не указан, по адресу в заголовке файла. Формат записи идентичен БК-0010, но для программ с нее нужно задавать имя (на запрос NAME) в виде: V:⟨имя⟩
- (область); U — вывести в файл на магнитофон содержимое памяти. ⟨Область⟩ — конструкция вида: ⟨хххххх⟩ — ⟨уууууу⟩ (хххххх — начальный адрес, уууууу — конечный адрес) или ⟨хххххх⟩ + ⟨уууууу⟩ (здесь уууууу — длина в байтах)
- ⟨1/0⟩ M — включить/выключить двигатель магнитофона
- ⟨хххххх⟩ G — запуск с адреса хххххх, который при этом помещается в ячейку 40 (USERPC)
- P — продолжить выполнение с места останова по СТОП
- ШАГ — выполнить одну команду и вернуться в Монитор

В мониторе также действуют все управляющие коды — клавиши сдвижки — раздвижки, «стрелки» и т. п.

Системные регистры

Изменения в системных регистрах БК-0011 по сравнению с БК-0010 следующие:

1) регистр данных клавиатуры (@ #177662): добавлены биты, доступные для записи:

15 — разрешение прерывания по СТОП (0-разрешение)

14 — прерывание от встроенного таймера 48,5 Гц

8—11 — номер палитры (по таблице)

2) системный регистр (@ #177716). Добавлены биты:

по чтению

8—15 — адрес начального пуска (140000)

7 — наличие арифметического сопроцессора (1/0 — нет/есть)

2 — признак записи в системный регистр по записи

11 — размещение записи в системный регистр

12—14 и 8—10 — номера страниц, подключаемых в адреса 40000—100000 и 100000—140000 (по таблице)

7 — управление двигателем магнитофона (1/0 — стоп/пуск)

0, 1, 3 и 4 — разрешение работы ПЗУ 8, 9, 10 и 11 (соответствующие страницы ОЗУ отключаются)

2 — подключение буфера экрана (1/0 — экран1/экран0)

Примечание: экранным буферам 1 и 2 соответствуют страницы ОЗУ 5 и 6.

3) Регистры дисководов:

@ #177130 — регистр состояния;

@ #177132 — регистр данных.

4) регистр принтера: при подключенном к порту (@ #177714) блоке контроллера принтера регистр порта преобразуется в регистр принтера и имеет формат:

бит 14 — бит строка;

бит 0—7 — байт данных.

Векторы прерываний

В БК-0011 предусмотрены следующие адреса прерываний:

4 — прерывание по СТОП;

10 — резервная команда;

14 — прерывание по T-разряду;

20 — прерывание по IOT;

24 — прерывание по аварии питания;

30 — прерывание по EMT;

34 — прерывание по TRAP;

60 — прерывание по верхнему регистру клавиатуры;

100 — прерывание от встроенного таймера;

270 — прерывание пользователя;

274 — прерывание по нижнему регистру клавиатуры;

300 — от блока контроллера локальной сети.

Адреса 40—56 и 64—66 зарезервированы для совместимости с БК-0010.

Системные драйверы

В БК-0011 введено большое число новых ЕМТ-функций. Одни из них выполняют необходимые при постраничной организации памяти действия по перезаписи данных из страницы в страницу и вызову подпрограмм из скрытых страниц, другие позволяют разнообразить оформление программ (вывод звука, рисование окружностей), третьи позволяют программировать перехват прерываний и ошибок, не вдаваясь в подробности описания системы и векторов прерывания.

Кроме того, улучшились возможности ЕМТ-функций, имеющихся в БК-0010. Так теперь данные для ЕМТ-функций передаются не через регистры, как было в БК-0010, а через стек. Только регистр R0 по-прежнему служит для передачи отдельных данных.

ЕМТ-функции, которыми может пользоваться программист БК-0011:

- ЕМТ0 — выход из программы пользователя в Монитор;
- ЕМТ10 — чтение строки с клавиатуры с возможностью экранного редактирования (можно вводить в состав строк и управляющие символы, как в режиме РЕД Фокала БК-0010);
- ЕМТ40 — ЕМТ50 — зарезервированы для совместимости с БК-0010;
- ЕМТ52 — подключение страницы с заданным номером;
- ЕМТ54 — вызов подпрограммы из скрытой страницы (JSR);
- ЕМТ56 — переход на скрытую страницу (JMP);
- ЕМТ60 — чтение из скрытой страницы;
- ЕМТ62 — запись байта в скрытую страницу;
- ЕМТ64 — установка параметров драйвера монитора (обращение к данной функции заменяет последовательности управляющих символов типа ИНВ, ПОДЧ и т. п.);
- ЕМТ66 — экранное редактирование уже имеющейся строки, для чего строка выводится на экран;
- ЕМТ72 — чтение статуса страницы (подключена или не

подключена; если подключена, то по какому адресу);

- ЕМТ74 — переключение буферов экрана (указание показываемого буфера и буфера для вывода информации);
- ЕМТ100 — установка режима клавиатуры (указание таблицы кодировки, отключение «щелчка» при нажатии, задание адреса подпрограммы обработки прерывания от клавиатуры);
- ЕМТ102 — ввод кода символа с клавиатуры без ожидания (INKEY);
- ЕМТ104 — чтение цвета экрана;
- ЕМТ110 — построчный сдвиг экрана по вертикали (SCROLLING);
- ЕМТ112 — перехват вектора прерывания @ # 4;
- ЕМТ114 — перехват ошибки «недопустимое ЕМТ»;
- ЕМТ116 — чтение состояния драйвера клавиатуры;
- ЕМТ120 — запись слова в скрытую страницу;
- ЕМТ124 — выдача звука с заданной длительностью и частотой;
- ЕМТ130 — рисование дуг и окружностей.

Кроме того, есть возможность вызова отдельных системных подпрограмм обычным путем — с помощью оператора JSR:

- 160000 — автоматическая загрузка ОС БК-0011 с диска (ОС сначала отыскивается на диске 0, а затем на дисках 1, 2 и 3);
- @ # 160002 — загрузка данных с заданного диска (0, 1, 2 или 3);
- @ # 160004 — чтение/запись по номеру логического блока;
- @ # 160006 — то же с явным указанием дорожки и сектора;
- @ # 160010 — инициализация рабочей области (буфера 60 байт, значение которого аналогично назначению буфера для ЕМТ36);
- @ # 160012 — форматирование заданной дорожки.

Бейсик

Версия Бейсика, зашитая в ПЗУ БК-0011, почти полностью аналогична версии Бейсика БК-0010. Отличия состоят

лишь в добавлении нескольких полезных операторов и функций, в том числе тех, которые не были реализованы на БК-0010, в изменениях, связанных с постраничной организацией памяти и в переименовании некоторых операторов «на заграничный лад» — для совместимой версии MSX.

Рассмотрим основные отличия Бейсика БК-0010 от Бейсика БК-0011:

1) в связи с увеличением объема памяти, отпущенного под транслятор, появилась возможность вывода расширенных сообщений об ошибке. Теперь вместо номера ошибки выводится краткая текстовая подсказка, как это сделано в Фокале БК-0010;

2) команда MONIT переименована в SYSTEM;

3) в связи с большим объемом памяти и с возможностью установки нескольких пользовательских ПЗУ появилась возможность использовать команду CALL, которая в Бейсике БК-0010 присутствует явно больше «для проформы»;

4) добавлена весьма полезная текстовая функция INSTR — поиск начала вхождения подстроки в текстовую строку;

5) в операторах DEF USR, BLOAD, BSAVE, POKE, PEEK, OUT и INP в список аргументов добавлено указание номера страницы ОЗУ;

6) добавлена команда SCREEN, позволяющая переключать буферы экрана, указывая номер экрана для вывода информации и номер экрана, демонстрируемого на мониторе (пример: SCREEN 1 SHOW 0), а также разрешающая/запрещающая переключение страницы ОЗУ к экрану (SCREEN 1 ON — разрешение, SCREEN 0 OFF — запрет); страницы ОЗУ, отключенные от экрана, могут быть использованы для программы пользователя;

7) улучшены возможности оператора BEEP — теперь можно задавать длительность и частоту звука;

8) реализована функция SPS, генерирующая указанное число пробелов при выводе на экран или принтер;

9) реализован оператор MERGE, позволяющий формировать программы на Бейсике из стандартных блоков, записанных на магнитофон;

10) появилась очень полезная при вызове USR-подпрограмм функция VARPTR, позволяющая определить адрес содержимого указанной переменной в ОЗУ БК, что позволяет передавать большее количество данных в подпрограмму USR.

11) появилась новая возможность в операторе DIM: если размерность массива записывать не в круглых, а в квадратных скобках, то будет создан виртуальный мас-

сив, расположенный в скрытых страницах. Это позволит хранить большие объемы данных, но время обращения к такому массиву больше, чем к обычному.

К сожалению, создатели БК-0011 приготовили пользователям этого компьютера неприятный сюрприз: в отдельных версиях Бейсика не реализованы операторы TRON и TROFF, рисование прямоугольников оператором LINE с аргументом B и WF и, что особенно плохо, не реализован оператор DRAW. Кроме того, время компиляции оказалось в 4—5 раз большим, чем в БК-0010. Это связано с тем, что Бейсик при компиляции постоянно переключает страницы, размещая данные в памяти наиболее «удобным» способом, а также хранит часть данных в скрытых страницах ОЗУ с большим временем доступа.

ОС БК-0011

91

Владельцы БК-0011 в комплекте с дисковыми программами могут воспользоваться возможностями операционной системы ОС БК-0011. Использование ОС в Бейсике позволяет дополнить стандартную версию Бейсика некоторыми полезными утилитами. При запуске Бейсика в среде ОС БК-0011 на экран один за другим выводятся вопросы: включать или не включать ту или иную утилиту в набор функций Бейсика. Это позволяет набрать необходимый набор функций, не расходуя память на остальные функции, не столь нужные. В качестве примеров функций операционной системы ОС БК-0011 можно привести следующие:

1) комбинация клавиш СЦ + С позволяет прерывать выполнение программы на Бейсике. Двухкратное нажатие вызывает немедленное прерывание, а однократное срывает «чуть позже», когда в программе встречается оператор ввода данных. Отдельная команда позволяет разрешить или блокировать данное использование комбинации СЦ + С;

2) значение устройства (DEVISE) позволяет установить устройство — адресат для передачи информации (терминал, дисквод или принтер);

3) установка длины телевизионной строки;

4) команда ABORT — оператор Бейсика, аналогичный NEW, но в отличие от него, позволяющий указать БК, нужно ли стереть данную программу из памяти, или ее следует оставить;

5) разрешение/запрет паузы по СЦ + О — аналог применяемой в БК-0010 комбинации СУ + @ и клавиши PAUSE IBM-подобных ЭВМ;

6) в набор Бейсика могут быть включены трансцендентные арифметические функции SIN, COS, EXP и т. д.

Запуск программ

в машинных кодах для БК-0010 и БК-0010.01 на БК-0011

Практически любая программа в машинных кодах, работающая на БК-0010 или БК-0010.01 (кроме некоторых программ, в которых применены нестандартные приемы программирования) может быть запущена на выполнение на БК-0011. Для этого нужно считать в ОЗУ БК-0011 с магнитной ленты заранее записанный туда Монитор БК-0010 (с незначительными изменениями, например, если вы захотите при работе с программами пользоваться исключительно дисковыми БК-0011, нужно заменить адрес подпрограммы — драйвера магнитофона для ЕМТ36 в таблице адресов ЕМТ на адрес драйвера дискового). Записать Монитор БК-0010 на магнитную ленту можно на БК-0010 или БК-0010.01, например, с помощью команды МЗ режима ТС (указав адрес начала для записи 100000₈ и соответствующую длину). После этого записанный Монитор загружается в ОЗУ БК-0011 командой L ee Монитора (имя задается с параметром В:) и при необходимости дорабатывается. После этого нужно запустить Монитор БК-0010 на исполнение как обычную программу в кодах для БК-0011 и тогда БК-0011... превращается в БК-0010, ибо они полностью совместимы по командам процессора, и разница между ними только в программной поддержке. Далее можно работать с БК-0011 точно так же, как и с БК-0010. Кстати говоря, в новой модификации БК-0011М этот недостаток уже устранен, и БК-0011М программно совместима с БК-0010 уже с самого начала.

Приложения

1. Таблица кодировки параметров страниц ОЗУ

бит:	14 (10)	13 (9)	12 (8)	номер страницы
1	1	0	0	страница 0 (0—40 000)
0	0	0	0	страница 1 100 000
0	1	0	0	страница 2 140 000
0	1	1	0	страница 3 180 000
1	0	0	0	страница 4 240 000
0	0	1	0	страница 5 (экран 0)
1	1	1	0	страница 6 (экран 1)
1	0	1	0	страница 7 (системная)

Страницы 8 и 9 ПЗУ зарезервированы под Бейсик (страница 8 и адреса 100000-117777 страницы 9) и драйвер-мониторную систему.

2. Кодирование символов в БК-0011.

БК-0011 позволяет работать с различными таблицами символов КОИ8, КОИ7—НО, КОИ7—Н1 и смешанной таблицей КОИ. Имеется также отдельная таблица для символов полуграфики, приближенных к прорисовке символов на IBM. Обе таблицы независимо друг от друга могут быть программно заменены на таблицы пользователя. Символы с кодами &НО—&Н20 и &НС0—&НFF соответствуют символам с этими кодами на БК-0010, а символы полуграфики (коды &НАО—&НBF) получили новую прорисовку (рис. 2).

Символы полуграфики БК-0011 и их шестнадцатичные коды

	⊕	⊖	▽	◆	♣	♠	⊕	⊖	⊙	♂	♀	♪	♫	✳	
	A0	A1	A2	A3	A4	A5	A6	A7	A8	AA	AB	AC	AD	AE	AF
К	→	↕		π	5	=	↕	↑	↓	→	←	↔	↕	↕	
00	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF

Рис. 2

3. Цветовые палитры БК-0011.

номер палитры	код точки			
	00	01	10	11
0	зеленый	синий	красный	
1	сиреневый	желтый	красный	
2	синий	голубой	сиреневый	
3	голубой	зеленый	желтый	
4	голубой	сиреневый	белый	
5	белый	белый	белый	
6	красно-коричневый	темно-красный	красный	
7	светло-зеленый	салатовый	желтый	
8	фиолетово-синий	фиолетовый	сиреневый	
9	фиолетово-синий	светло-зеленый	красно-коричневый	
10	фиолетовый	салатовый	темно-красный	
11	желтый	голубой	красный	
12	зеленый	красный	голубой	
13	желтый	голубой	белый	
14	зеленый	желтый	белый	
15	зеленый	голубой	белый	

Знакомьтесь: БК-0011М

В компьютере БК-0011М оказалось много недостатков, самыми крупными из которых обладала ДМС. Поэтому были начаты работы по его модернизации. В результате скоординированных усилий разработчиков программного и аппаратного обеспечения был создан новый компьютер, который получил название БК-0011М.

В аппаратную часть БК-0011М введено раздельное управление памятью и магнитофоном, добавлены устройство управления палитрами, блокировка клавиши «СТОП» и некоторые другие изменения. Наиболее крупным достоинством БК-0011М служит системное ПЗУ, которое получило название «Базовая операционная система» (БОС). Возможности БОС по сравнению с ДМС намного шире: существенно увеличена скорость вывода на экран, расширен набор графических примитивов и системных функций, добавлен режим вывода 80 знаков в строке, управляющие коды экрана и клавиатуры совместимы с терминалом VT-52 и ЭВМ серии ДВК с дополнительными возможностями инверсии, подчеркивания, управления цветами, палитрами, шириной и высотой символов, сдвижки/раздвижки строки и экрана. Бейсик БК-0011М стал легко совместимым с БК-0010 с добавлением новых возможностей и исправлением некоторых ошибок.

Эти программные и аппаратные решения сделали возможным использование программ, разработанных как для БК-0010, так и для ДВК и УКНЦ.

Эволюция операционной системы

В основу ОС БК-11 легла популярная операционная система RT-11 фирмы DEC. Эта ОС известна у нас пользователям ЭВМ серии СМ, ДВК, «Электроника» под названиями РАФОС, ФОДОС и др. Однако исходная версия RT-11 не могла работать на БК-0011 из-за отсутствия регистров вывода на терминал, нестандартных регистров клавиатуры и неработоспособности драйвера НГМД в составе ДМС. Поэтому потребовалась разработка нового драйвера нижнего уровня, помещенного в ПЗУ контроллера НГМД, системного драйвера для RT-11 и адаптация самой системы. На базе версии 5.2 RT-11 была создана ОС БК-11 версии 1.0. Драйвер НГМД ВУ по формату дискеты был сделан совместимым с драйвером МУ от

ДВК и драйвером MZ УКНЦ. С появлением БК-0011М версия 1.0 была переделана в 1.1. Была также создана новая версия драйвера для ПЗУ контроллера НГМД. Версия 1.1 имела один существенный недостаток: ввод с клавиатуры производился, минуя БОС, что делало коды функциональных клавиш несовместимыми с ДВК.

В версии 2.1 этот недостаток был устранен. Кроме того, был ускорен вывод на экран символьных строк за счет использования соответствующей функции БОС. Эта версия имела ограниченное распространение.

В версии 2.3 были введены значительные усовершенствования: реализованы два режима работы — стандартный и виртуальный, в которых все части ОС и файл свопинга хранятся в скрытых страницах ОЗУ, что существенно сокращает число обращений к системному диску и ускоряет работу. Кроме того, был усовершенствован драйвер диска: введено отключение двигателя дисководов по таймеру, а также добавлен ряд SET-параметров для настройки драйвера на конкретный привод (количество дорожек, время перемещения с дорожки на дорожку, дорожка прекоррекции и др.). В драйвер были включены программы форматирования и проверки дискет, что позволило отказаться от программы TESTBU. Также были добавлены драйверы для настройки режимов вывода на экран и ввода с клавиатуры.

Наконец, в 1991 г. была создана ОС БК-11 версии 4.0 на базе версии 5.4 RT-11. Версия 4.0 унаследовала все нововведения 2.3, был добавлен автоповтор клавиатуры, устранены ошибки и усовершенствованы некоторые драйверы. От RT-11 был унаследован новый символьный отладчик, позволяющий работать с ассемблерными мнемониками и таблицей символов пользовательской программы, и добавлены новые возможности для разработки драйверов. Распространение и поддержку ОС БК-11 версии 4.0 осуществляет НПКП «ИнтерСервер».

Обзор системного и прикладного ПО

Программная совместимость ОС и формата гибких дисков позволяет использовать все программное обеспечение, разработанное для ДВК и УКНЦ. Ограничения, накладываемые на программы:

программа должна работать в текстовом

режиме, пользуясь средствами ОС, и не обращаться к регистрам внешних устройств;

для управления экраном и распознавания функциональных клавиш программа должна использовать управляющие последовательности терминала VT-52 фирмы DEC. (Заметим, что с этим терминалом совместимы дисплей 15-ИЭ-00-13, контроллеры КСМ и КЦГД в одном из режимов, а также УКНЦ.);

программа не должна быть слишком большой, так как объем одновременно доступного ОЗУ в БК-0011М на 8 Кбайт меньше.

Этим ограничениям удовлетворяют почти все системы программирования (MACRO, C, PASCAL, FORTRAN, Modula-2), экранные редакторы (EDIK, K52, SCREEN) и многие другие программы.

Рассмотрим далее некоторые особенности программ, специально разработанных для БК-0011М.

94

Операционная оболочка Inter Commander

Аналог популярной оболочки MS-DOS Norton Commander. Позволяет в простой и удобной форме, не используя команд ОС, просматривать каталоги устройств, запускать, редактировать, копировать, переименовывать и удалять файлы и группы файлов, а также вводить, редактировать и выполнять команды операционной системы. Кроме того, можно настроить функции просмотра, редактирования и запуска в зависимости от типа файла. Программа имеет встроенную подсказку с описанием функциональных клавиш. Каталоги программ в панелях могут быть отсортированы по различным параметрам. Имеется иерархическое меню, настраиваемое пользователем. Работа с программой возможна с использованием клавиатуры или манипулятора «мышь» (УВК-01).

Экранный редактор EdikM

Аналог известного на ДБК редактора EDIK. Помимо полного набора функций EDIK'a, обладает следующими возможностями:

ускоренная работа с экраном за счет использования функций БОС;

переключение ширины символов внутри редактора;

автоматическое выравнивание при переходе на следующую строку;

введена дополнительная область текста, с которой можно производить операции выделения инверсным цветом, копирования, перемещения, удаления и записи во внеш-

ний файл без ограничения размера этой области;

ввод в текст символов псевдографики.

Исполняющая система Exe10plus

Позволяет эмулировать среду БК-0010 и дает возможность:

исполнения программ для БК-0010 в машинных кодах, на Фокале и Бейсике;

переназначения ввода/вывода с магнитного фона на диск;

загрузки исполняемых программ на место ПЗУ Фокала;

возврата в ОС с сохранением экрана и режимов работы ОС.

Программа PCBK

копирования дискет IBM PC

Программа позволяет считывать, записывать и инициализировать на БК-0011М дискеты в формате IBM PC. Пользовательский интерфейс программы выполнен в стиле Norton Commander. Программа поддерживает форматы 320К, 720К и 800К, используемые на IBM PC. При копировании может производиться перекодировка текстовых файлов из кодировки БК-0011М в альтернативную кодировку, принятую на IBM PC, и обратно. Имеется также и программа для IBM PC, позволяющая копировать дискеты БК-0011М.

Манипулятор «мышь» (УВК-01)

Драйвер манипулятора выполнен в виде драйвера ОС и имеет широкий набор функций для прикладных программ. Возможно использование его в режиме эмуляции клавиатуры (например, в экранных редакторах).

Графический редактор Paint11M

Это развитие известного на БК-0010 редактора Paint (авт. Бакерин). Работает под управлением ОС БК-11. Созданный рисунок может записываться и считываться с диска, а также печататься на Epson-совместимых принтерах, работающих в стандартах ИРПР и Centronics (например, Robotron или MC-6313). Достигнута идентичность изображения рисунка на экране рисунку на листе бумаги, увеличены размеры перемещаемых и копируемых областей.

Программа просмотра баз данных

Программа позволяет просматривать файлы баз данных типа DBF (dBASE, FoxBase, Clipper и др.). Предусмотрены режимы просмотра по записям и в виде таблицы, выдача справочной информации о базе данных и полях, переход по номеру записи, поиск подстроки, вывод на печать всей базы или выбранных записей.

Заканчивается разработка редактора баз данных.

Программа печати текстов Kprint

Kprint позволяет устанавливать режим и шрифт печати любого Epson-совместимого принтера, выбирать файл для печати в режиме меню. Возможно также динамическое управление режимами и шрифтами благодаря набору управляющих символов, встраиваемых в текст.

Библиотеки графики и функций БОС

Объектные библиотеки для языков программирования Си и Паскаль предоставляют набор функций для доступа к возможностям БОС по выводу на экран текстовой и графической информации, управления расширенной памятью и др.

Сетевые возможности

Отличительной чертой компьютеров БК-0011 и БК-0011М является наличие в их аппаратуре контроллеров локальной сети с достаточно высокими характеристиками, что выгодно отличает их от других компьютеров аналогичного класса.

Первоначально для БК-0011 и БК-11М был предложен контроллер локальной сети (КЛС), аналогичный контроллеру ПЭВМ УКНЦ. В дальнейшем не нашел применения из-за отсутствия программного обеспечения.

Другим решением стал контроллер моноканальной сети (КМК). Благодаря этому сеть на базе компьютеров БК-0011 и БК-0011М обладает низкой себестоимостью, достаточной надежностью и высокими техническими характеристиками. Проводящий средой служит витая пара проводов, дальность связи — до 300 м, скорость передачи данных — 75 КБод. Одновременно в сети могут работать до 30 компьютеров. Программное обеспечение нижнего уровня выполнено по модели OSI и прошито в ПЗУ, расположенном в дополнительных страницах памяти. Им обеспечивается обмен данными между любыми

абонентами сети в широкоэвентальном и избирательном режимах.

Штатное программное обеспечение более высокого уровня не позволяет использовать преимущества этой сети. Поставляемая заводом-изготовителем программа DIALOG не обладает удобным современным интерфейсом для пользователя, имеет ограниченное число функций, не позволяют просто расширять систему. Кроме того, весьма неудачна работа в сети Бейсик-системы, приведшая к несовместимости к КУВТ-86, и кроме того, невозможно получать тексты программ и файлы данных на центральной ПЭВМ. DIALOG не обеспечивает выполнения на периферийных станциях программ для БК-0010 и программ операционной системы RT-11, что сильно ограничивает область применения сети.

Перечисленные недостатки инициировали создание более совершенного программного обеспечения.

Для сети разработана система Exel10Net для класса на базе БК-0011. Она позволяла выполнять на периферийных ПЭВМ программы БК-0010 в машинных кодах, а также запускать интерпретатор языка Focal. Система не позволяла вести обмен данными, но тем не менее скрасила практически полное отсутствие программ для БК-0011.

Затем система Exel10Net была перенесена на класс УКНЦ-01.01, и дальнейшее развитие систем этой линии происходило исключительно на БК-0011М. С введением обмена данными периферийных ПЭВМ с дисководом центральной машины появилась сетевая система SPRUT1. Простота работы в режиме меню и обмен данными позволяли применять в классе широко распространенные программы для БК-0010.

Следующим шагом стала разработка принципиально новой по возможностям и интерфейсу системы SPRUT2. В первую очередь была введена возможность выполнения на периферийных ПЭВМ программ самого БК-0011М. Совершенно изменен интерфейс пользователя: работа ведется в двухпанельном меню в стиле Norton Commander, что в корне отличает систему от подобных ей и делает работу очень удобной, отвечающей современным требованиям. Система SPRUT2 поддерживает работу в сети загружаемого Бейсика, совместимого с Бейсиком БК-0010, который обеспечивает обмен текстами программ и файлами данных с диском центральной ПЭВМ. Кроме того, в сети выполняются все программы БК-0010 в машинных кодах и на Фокале.

Весьма важно, что сетевая система SPRUT2 рассчитана на работу в качестве управляющей системы в сети на базе компьютеров БК-0011М с центральной ПЭВМ типа

IBM PC. В системе также предусмотрено значительное расширение выполняемых функций и задач сети без потери совместимости с предыдущими версиями системы.

Наконец, недавно появившаяся система SPRUT3 создает на периферийных компьютерах среду операционной системы ОС БК-11 и позволяет работать со всем спектром программ для БК-0010 и БК-0011М (например, с редакторами Lexis, EDASP, EdikM, Paint11M и др.), а также с большим количеством программ от УКНЦ и ДВК (в том числе с трансляторами MACRO, PASCAL, C, FORTRAN).

Перспективы развития

Какую же нишу может занять БК-0011М в отечественном компьютерном рынке? Она определяется его относительно невысокой ценой и достаточно большими профессиональными возможностями. Перечислим области наиболее рационального использования этой ЭВМ.

В малом бизнесе как в качестве персональной ЭВМ, так и в качестве станции в составе локальной сети с IBM PC. Успех в этой сфере будет определяться количеством и качеством прикладного программного обеспечения: текстовых процессоров, электронных таблиц, баз данных и др.

В составе учебных компьютерных классов. Успех в этой сфере опеределяется возможностью использования большого количества учебных и игровых программ, разработанных для БК-0010 и КУВТ-86. Начинают появляться и учебные программы и системы, созданные специально для классов БК-0011М. Применение в качестве центральной ЭВМ в классе IBM PC повышает надежность работы класса и дает возмож-

ность постепенного перехода обучения школьников на новый уровень.

Использование копировщика РСВК позволяет обмениваться текстовыми файлами и базами данных с IBM PC. В перспективе подключение к телефонной сети через модем даст возможность связи с другими БК-0011М, IBM PC, электронными досками объявлений и глобальными системами компьютерной связи.

НПКП «ИнтерСервер» продолжает работу над усовершенствованием программ для БК-0011М и разработкой новых. В программе работ новые версии операционной системы, оболочки InterCommander, программы РСВК и других. Будут продолжены работы по развитию локальной сети IBM PC — БК-0011М. Ведутся работы по подключению модема и «винчестерского» диска.

Важным направлением является разработка системы управления базами данных, совместимой по файлам с dBASE, для дисковой и сетевой конфигураций. В сетевой конфигурации СУБД будет функционировать по модели «клиент — сервер» с прямым доступом к файлам баз данных на жестком диске IBM PC, что позволит оптимально распределить функции между IBM PC и БК-0011М, снизить нагрузку на сеть и в конечном счете достичь высокой эффективности многопользовательской работы. Реализация прикладных задач на основе сетевой базы данных позволит создавать недорогие многопользовательские АРМы для применения в малом бизнесе, на складах, в магазинах и пр.

«ИнтерСервер» приглашает к сотрудничеству разработчиков программного обеспечения и аппаратных средств для объединения усилий по расширению области применения БК-0011М.

А. КОНЮШЕНКО

(г. Минск)

Система машинных команд БК-0010

Рассказывать о недостатках БК-0010 можно очень долго. К этим недостаткам относится и никуда негодная документация, хотя у большинства владельцев и такой нет. Создается впечатление, что программисты, пишущие программы для БК, не желают раскрывать свои профессиональные тайны широкому кругу владельцев этих машин. Например, известный комплект программ Ю. Зальцмана СИСТЕМА долгие годы ходил в кассетном варианте и только в прошлом году наконец появился в печати. Существуют и другие комплекты программ аналогично-

го назначения, путь которых к рядовым пользователям менее удачлив.

Однако не все так просто. Дело в том, что у пишущего программы не остается свободного времени на рассказы о том, как он это делает. У обучающих программистов нет времени на исследование «уникальных» аппаратных особенностей БК... Но, даже написав более-менее удовлетворительное справочное пособие, весьма проблематично довести его до издания в приемлемые сроки...

Программисты-любители, «ковыряющие»

А. Кошманко

= UC2AAU =

220030 Минск-30. я. 1

БЕЛАРУСЬ

КОДЫ МАШИНЫХ КОМАНД									
	1801BM1 (PDP-11)	00 4R DD	JSR	10 00 00 +XXX	BPL				
		00 50 DD	CLR	10 04 00 +XXX	BMI				
00 00 00	HALT	00 51 DD	COM	10 10 00 +XXX	BHI				
00 00 01	WAIT	00 52 DD	INC	10 14 00 +XXX	BLOS				
00 00 02	RTI	00 53 DD	DEC	10 20 00 +XXX	BVC				
00 00 03	BPT	00 54 DD	NEG	10 24 00 +XXX	BVS				
00 00 04	IOT	00 55 DD	ADC	10 30 00 +XXX	BCC, BHIS				
00 00 05	RESET	00 56 DD	SBC	10 34 00 +XXX	BCS, BLO				
00 00 06	RTT	00 57 DD	TST	10 40 00 - 10 43 77	BMT				
		00 60 DD	ROR	10 44 00 - 10 47 77	TRAP				
(00 00 07 - 00 00 11) [1a]		00 61 DD	ROL	10 50 DD	CLRB				
		00 62 DD	ASR	10 51 DD	COMB				
00 00 12	START	00 63 DD	ASL	10 52 DD	INCB				
		00 64 NN	MARK	10 53 DD	DECB				
(00 00 13 - 00 00 15) [1b]				10 54 DD	NEGB				
		(00 65 SS)	(MFPI) [2]	10 55 DD	ADCB				
00 00 16	S	(00 66 DD)	(MTPI) [3]	10 56 DD	SBCB				
				10 57 DD	TSTB				
(00 00 17 - 00 00 77) [1c]		00 67 DD	SXT	10 60 DD	RORE				
				10 61 DD	ROLB				
00 01 DD	JMP	(00 70 00 - 00 77 77)[4]		10 62 DD	ASRB				
00 02 0R	RTS			10 63 DD	ASLB				
		01 SS DD	MOV	10 64 SS	MTPS				
(00 02 1R тех. контроль)		02 SS DD	CMP						
(00 02 2N тех. контроль)		03 SS DD	BIT	(10 65 SS)	(MFPD) [9]				
(00 02 3N) (SPL)		04 SS DD	BIC	(10 66 DD)	(MTPD) [10]				
		05 SS DD	BIS						
00 02 40	NOP	06 SS DD	ADD	10 67 DD	MFPS				
00 02 41 - I операции с									
- 00 02 77 Iразрядами PSW		(07 0R SS)	(MUL) [5]	(10 70 00 - 10 77 77) [11]					
00 03 DD	SWAB	(07 1R SS - 07 3R SS)[6]							
00 04 00 +XXX	BR			11 SS DD	MOVE				
00 10 00 +XXX	BNE	07 4R SS	XOR	12 SS DD	CMPB				
00 14 00 +XXX	BEQ			13 SS DD	BITB				
00 20 00 +XXX	BGE	(07 50 0R - 07 50 3R)[7]		14 SS DD	BICB				
00 24 00 +XXX	BLT	(07 50 40 - 07 67 77)[8]		15 SS DD	BISB				
00 30 00 +XXX	BGT			16 SS DD	SUB				
00 34 00 +XXX	BLE	07 7R NN	SOB	(17 00 00 - 17 77 77) [12]					

Область [1а-с] в системе команд PDP-11 не используется; у процессора 1801BM1 в этой области имеются команды START и S (шаг), но вследствие отсутствия системных регистров, БК-0010(01) по этим командам обрабатывает вектор 4.

[2-3] команды процессоров с MMU

[4, 8, 11] резервный код

[5] команда умножения (реализована в 1801BM1Г)

[6] расширенный набор команд арифметики

[7, 12] команды процессора с плавающей точкой

[9, 10] команды процессоров с 22-разрядным MMU

UC2AAU - БК-0010 - Справочное пособие

В кодах машинных команд	В обозначениях операций
* = 0 для операций над словами 1 для операций над байтами	(N) = содержимое ячейки или рег. N
SS = поле адресации операнда-источника	s = операнд-источник (scr)
DD = поле адресации операнда-приемника	d = операнд-приемник (dat)
R = PОН N 0...7 (3 бита)	r = содержимое регистра
XXX = смещение (-128, ..., +127; 8 бит)	<= = становится равным
N = число, 3 бита	X = относительный адрес
NN = число, 6 бит	% = определение регистра
ЛОГИЧЕСКИХ ОПЕРАЦИЙ	ОПЕРАЦИИ НАД РАЗРЯДАМИ PSW
∧ = логическое И	* = установка/сброс по результату
∨ = логическое ИЛИ	- = состояние разряда не меняется
\\ = исключающее ИЛИ	0 = сброс
⌊ = НЕ	1 = установка

МЕТОДЫ АДРЕСАЦИИ

	Имя-моника	МЕТОД	R
0 регистровая	R	(R)	- операнд
1 косвенная регистровая	(R) или @R	(R)	- адрес операнда
2 автоинкрементная	(R)+	(R)	- исполняемый адрес (R) <= (R)+2 (или 1)
3 косв. автоинкрементная	@(R)+	(R)	- адрес адреса (R) <= (R)+2 (или 1)
4 автодекрементная	-(R)	(R) <= (R)-2 (или 1)	до исп. ком. (R) - исп. адрес
5 косв. автодекрементная	@-(R)	(R) <= (R)-2 (или 1)	до исп. ком. (R) - адрес адреса
6 индексная	X(R)	(R)+X	- адрес
7 косв. индексная	@X(R)	(R)+X	- адрес адреса

С ИСПОЛЬЗОВАНИЕМ СЧЕТЧИКА КОМАНД

	МЕТОД	7
2 непосредственная	#n	операнд n в следующем за командой слове
3 абсолютная	@#A	адрес A в следующем за командой слове
6 относительная	A	адр. операнда = адр. ком. +4+X (+6+X)
7 косв. относительная	@A	адр. адреса опер. = адр. ком. +4+X (+6+X)

ВЕКТОРЫ ПЕРЕРЫВАНИЯ

Источник прерывания	Адрес вектора
Зависание при передаче данных по каналу ИЛИ от клавиши <СТОП>	000004
Резервный код команд	000010
Прерывание по T-разряду	000014
Прерывание по команде IOT	000023
Авария сетевого питания (аппаратно не поддержан)	000024
Прерывание по команде EMT	000030
Прерывание по команде TRAP	000034
Прерывание от клавиатуры	000060
Сигнал IRQ2	000100
Прерывание от клавиатуры (коды нижнего регистра)	000274

РАБОТА С ПОДПРОГРАММАМИ

00 00 00	HALT	останов (в БК-0010 при попытке записи PSW в несуществ. регистр 177676, происх. прер.4)	(177716) <= 000010 \ (177716) (177676) <= PSW <= (160004) PC <= (160002)
00 00 01	WAIT	пауза - ожидание прерывания	
00 00 02	RTI	возврат из прерывания	PC <= (SP)+ PSW <= (SP)+
00 00 03	BPT	отладочное прерывание	-(SP) <= PSW <= (16)
		вектор 14	-(SP) <= PC <= (14)
00 00 04	IOT	вызов системы ввода-вывода	-(SP) <= PSW <= (22)
		вектор 20	-(SP) <= PC <= (20)
00 00 05	RRESET	сброс магистрали и процессора	
00 00 06	RTT	возврат, с запретом прерывания по T-разряду до исп. сл. команды	PC <= (SP)+ PSW <= (SP)+
00 01 DD	JMP	безусловный переход	PC <= d
00 02 0R	RTS	возврат из подпрограммы	PC <= R <= (SP)+
00 02 40	NOP	нет операции (пустая команда)	
00 4R DD	JSR	вызов подпрограммы	-(SP) <= R <= PC <= d SP <= PC + (2 x NN)
00 64 NN	MARK	восстановление стека	PC <= R5 <= (SP)+
07 7R NN	SOB	выч.1 и ветвл., если (R#) не 0	R# <= R# - 1 PC <= PC - (2 x NN)
10 40 00 -	EMT	вызов ОС, вектор 30	-(SP) <= PSW <= (32)
- 10 42 77			-(SP) <= PC <= (30)
10 44 00 -	TRAP	общего назн., вектор 34	-(SP) <= PSW <= (36)
- 10 47 77			-(SP) <= PC <= (34)
10 64 SS	MTPS	запись PSW	PSW <= a
10 67 DD	MFPS	чтение PSW	d <= PSW

99

ПЕРЕХОДЫ ПО УСЛОВИЮ (ВЕТВЛЕНИЯ): Вж* <адр.перехода>

	Базовый	КОП	1+/-	X X X		
	15		8	7	0	
			Если условие удовлетворяется, то (PC) <= (PC) + (2 x XXX)			
ВОСЬМЕРИЧНЫЕ	00 04 00	+XXX	BR	безусловный переход		
	00 10 00	+XXX	BNE	нет равенства (нулю)	Z = 0	
V = ТРИАДЫ	00 14 00	+XXX	BRQ	равенство (нулю)	Z = 1	
0 0 0 0	10 20 00	+XXX	BVC	арифм. переп. отсутствует	V = 0	
	10 24 00	+XXX	BVS	произошло арифм. переп.	V = 1	
1 0 0 1	10 30 00	+XXX	BCC	перенос отсутствует	C = 0	
	10 34 00	+XXX	BCS	произошел перенос	C = 1	
			Переход по знаку			
2 0 1 0	10 00 00	+XXX	BPL	знак плюс	N = 0	
	10 04 00	+XXX	BMI	знак минус	N = 1	
3 0 1 1	00 20 00	+XXX	BGE	больше или равно (нулю)	N \ V = 0	
	00 24 00	+XXX	BLT	меньше (нуля)	N \ V = 1	
4 1 0 0	00 30 00	+XXX	BGT	больше (нуля)	Z \ (N \ V) = 0	
	00 34 00	+XXX	BLE	меньше или равно (нулю)	Z \ (N \ V) = 1	
5 1 0 1			Переход без знака			
	10 10 00	+XXX	BHI	больше	C \ Z = 0	
6 1 1 0	10 14 00	+XXX	BLOS	меньше или равно	C \ Z = 1	
	10 30 00	+XXX	BHIS	больше или равно (= BCC)	C = 0	
7 1 1 1	10 34 00	+XXX	BLO	меньше (= BCS)	C = 1	

UC2AAU - БК-0010 - Справочное пособие

ОДНООПЕРАНДНЫЕ КОМАНДЫ: OPR dst

Код операции (КОП)		D D				N Z V C		
15	6	5	0					
00 03 DD	SWAB	перестановка байтов			**00			
*0 50 DD	CLR(B)	очистка	(d) <= 0		0100			
*0 51 DD	COM(B)	побитная инверсия	(d) <= (~d)		**01			
*0 52 DD	INC(B)	прибавление 1	(d) <= (d) + 1		***-			
*0 53 DD	DEC(B)	вычитание 1	(d) <= (d) - 1		***-			
*0 54 DD	NEG(B)	изменение знака	(d) <= -(d)		****			
*0 55 DD	ADC(B)	прибавить перенос	(d) <= (d) + C		****			
*0 56 DD	SBC(B)	вычесть перенос	(d) <= (d) - C		****			
*0 57 DD	TST(B)	проверка	(d) <= (d)		**00			
*0 60 DD	ROR(B)	цикл. сд. вправо	=> C,d		****			
*0 61 DD	ROL(B)	цикл. сд. влево	C,d <=		****			
*0 62 DD	ASR(B)	арифм. сд. вправо	(d) <= (d)/2		****			
*0 63 DD	ASL(B)	арифм. сд. влево	(d) <= 2(d)		****			
00 67 DD	SXT	расширить знак	N = 0, (d) <= 0 N = 1, (d) <= 177777		010-			

ДВУХОПЕРАНДНЫЕ КОМАНДЫ: OPR src dst; OPR src R; OPR R dst

КОП		S S		D D				N Z V C		
15	12	11	6	5	0					
*1 SS DD	MOV(B)	переслать	(d) <= (s)			**0-				
*2 SS DD	CMP(B)	сравнить	(s) - (d)			****				
*3 SS DD	BIT(B)	проверить разряды	(s)^(d)			**0-				
*4 SS DD	BIC(B)	очистить разряды	(d) <= (~s)^(d)			**0-				
*5 SS DD	BIS(B)	установить разряды	(d) <= (s)^(d)			**0-				
06 SS DD	ADD	сложить	(d) <= (s) + (d)			****				
07 4R SS	XOR	исключающее ИЛИ	(s) <= (r)^(s)			**0-				
16 SS DD	SUB	вычесть	(d) <= (d) - (s)			****				

ОПЕРАЦИИ С РАЗРЯДАМИ PSW:

Базовый КОП = 240		10/1	N	Z	V	C		
15	8	7	4	3	2	1	0	
ЛОГИЧЕСКИЕ ОПЕРАЦИИ:		Очистить						
	00 02 41	CLC	C	-	-	-	0	
	00 02 42	CLV	V	-	-	0	-	
(s)	00 02 44	CLZ	Z	-	0	-	-	
	00 02 50	CLN	N	0	-	-	-	
(d)	00 02 57	CCC	N Z V C	0	0	0	0	
		Установить						
(s)^(d)	00 02 61	SKC	C	-	-	-	1	
	00 02 62	SKV	V	-	-	1	-	
(s)^(d)	00 02 64	SKZ	Z	-	1	-	-	
	00 02 70	SEN	N	1	-	-	-	
(s)^(d)	00 02 77	SCC	N Z V C	1	1	1	1	

100

чужие программы в кодах или пишущие свои, часто сталкиваются с непонятным поведением машины. Поиски ответов на возникающие вопросы, как правило, не приводят к успеху, так как слишком велики различия между БК и ее «приблизительными» прототипами или «аналогами»: ДВК, «Электроника-60», PDP-11 и др.

Система машинных команд БК-0010 в более-менее компактном (табличном) виде охватывает всю систему команд PDP-11,

С. ТАЛАНОВ

(г. Яранск)

БК-0010 и телетайп

Кто из владельцев БК никогда не мечтал о принтере? Принтере, на котором можно напечатать все: от поздравительной открытки и письма к другу, до сложной радиосхемы и занимательной картинке? К сожалению, наша действительность такова, что мечтающих намного больше, чем владельцев этого замечательного устройства. В 1991 г. на прилавках магазинов «Электроника» в Москве и других городах можно было встретить относительно дешевые принтеры «Электроника MC6312» (термоструйная печатающая головка, стандарт EPSON LX-800) и «Электроника MC6313/EPSON». Принтер MC6313 особо удачная модель: матричная печать, полностью совместим со стандартом FX-800, широчайшие возможности печати текстов и графики.

Оба принтера при поддержке соответствующих драйверов без проблем работают с БК, как в текстовом, так и в графическом режиме. Но сейчас уже 1993 г., на прилавках «Электроники» пусто, куда пропали эти принтеры и сколько они сейчас стоят, не знает никто.

Чтобы в вопросах печати сколь-нибудь переместиться от «мечтающих» к «имеющим», автор предлагает вашему вниманию данную статью.

Как известно, на сегодняшний день, к БК подключено в качестве печатающих устройств немало различной «экзотики»: пишущие машинки «Консул», различные зарубежные принтеры, термопечатающие устройства, кассовые аппараты... Но девиз «Подключить все, что печатает» далеко не исчерпан. На предприятиях связи повсеместно используются телеграфные аппараты, работающие с кодом МТК-2 (международный телеграфный код). Это импортные аппараты Т-100, Т63, отечественные СТ-35, СТ67М и другие. В последнее время им на

в которой выделены команды процессора ВМІ. Далее приводятся сведения о командах, выполняемых БК. Команды условно разделены на пять групп по функциональному назначению. Дополнительно включены таблицы перевода восьмеричных чисел в триады, а также результаты основных логических операций.

Все замечания и уточнения редакция примет с благодарностью.

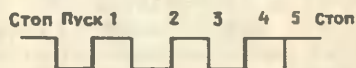
смену приходят электронные матричные телеграфные аппараты РТА-80 и F-2000. Телеграфные аппараты можно найти и на различных предприятиях, где они используются в качестве абонентских установок общесоюзной телетайпной сети.

Для использования в качестве печатающей установки необходима только приемная часть аппарата, без клавиатуры, трансмиттера, перфоратора. Я думаю, многим не составит большого труда приобрести приемную часть от списанного телеграфного аппарата на предприятиях связи.

При включении в сеть аппарата необходимо учесть, что некоторые из них работают только от 127 В, в этом случае их необходимо питать через понижающий трансформатор.

Код МТК-2 является равномерным, последовательным и состоит из сочетаний или чередований пяти токовых или бестоковых посылок. Для приема любого знака стартопным аппаратом требуется семь посылок — одна пусковая, затем 5 кодовых и одна стоповая. Все семь посылок вместе составляют стартопную комбинацию знака. Пусковая посылка в этой комбинации запускает в работу распределительный механизм приемной части аппарата, а стоповая посылка после каждого оборота механизма останавливает его. Пусковая и стоповая посылки называются коррекционными.

Общее число посылок кода МТК-2 составляет 32.



Пример кодирования: комбинация посылок для буквы «ы».

Как же научить БК работать с телетайпом? Для этой цели служит резидентный

драйвер, коды которого приведены ниже. Программа перекодирует код символа БК в код МТК-2 и выдаст его в последовательном коде на контакт А16 разъема порта пользователя (УП). При этом никакой доработки компьютера не требуется.

Однако приемный электромагнит телеграфного аппарата питается от внешнего источника питания напряжением 120 В, ток протекающий в цепи электромагнита, достигает 50 мА. Чтобы согласовать ТТЛ выход порта с питанием электромагнита, необходимо собрать схему согласования, в телеграфии такая схема называется однополюсным реле.

развязки выхода прерывателя от компьютера. Таким образом, однополюсное реле преобразует ТТЛ сигналы БК в сигналы с напряжением 0—120 В.

Для работы схемы необходим внешний источник постоянного напряжения 120 В с током до 1А. При полностью открытых ключах VT3, VT4, (режим СТОП) через электромагнит должен протекать ток 40—45 мА. Этот ток регулируется переменным резистором R19. При подключении источника необходимо соблюдать полярность. В противном случае выйдут из строя транзисторы VT3, VT4.

Машинные коды драйвера печати приведены в конце статьи. Их необходимо ввести в память компьютера в режиме МСТД. Автор опускает процесс ввода кодового модуля, так как он достаточно описан в различной литературе. Контрольная сумма программы, вычисленная также в режиме МСТД по директиве, X-013307.

Итак, вы ввели коды драйвера программы, собрали и подключили схему согласования, телеграфный аппарат. Приступаем к работе.

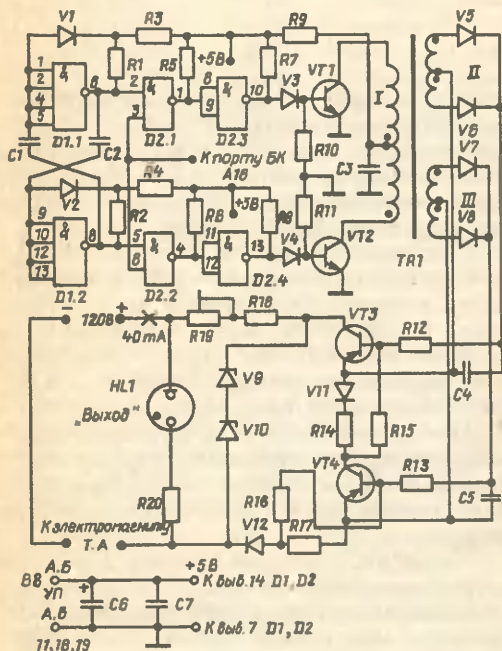
Драйвер печати может быть загружен с любого адреса в память, так как он является полностью перемещаемым, может работать в добавочном ОЗУ или ПЗУ пользователя. После запуска из пускового или отладочного мониторов программа выдает краткую инструкцию пользователю. Для выхода в режим МСТД или пусковой монитор необходимо нажать клавишу ВВОД. Для выхода в Фокал или Бейсик — ПРОБЕЛ. Следует отметить, что при работе с языками они должны быть сначала инициализированы — это относится только к тем машинам, которые имеют переключатели языков и МСТД, и может возникнуть, например, такая ситуация, что до включения машины в сеть язык был выключен, затем включен, но не запущен — в таком случае выхода в язык по директиве ПРОБЕЛ не произойдет, для этого нужно сначала запустить язык, а потом выйти из него директивой в монитор — из Бейсика-MON из Фокала-Р М.

После выхода в монитор или язык драйвер готов к работе, но себя пока никак не проявляет. Можно выполнять любые операции, грузить и запускать программы.

Данный принцип работы резидентного драйвера взят из программ печати драйверов принтеров автора Зальцмана Ю. А. (г. Алма-Ата).

Драйвер имеет два режима печати: чтобы вызвать режим печати, при котором будут печататься все знаки, выводимые на экран, необходимо нажать вместе клавиши AP2/ПУС; в служебной строке появится длинная черта — признак режима; выход — повторное нажатие клавиш AP2/ПУС. Другой режим обеспечивает прямую печать с клавиатуры: все символы, введенные пользова-

102



Принцип работы схемы:

ТТЛ сигналы поступают с контакта А16 порта БК на один из входов схем совпадения D2.1, D2.2. На другие входы этих схем поступают в противофазе импульсы высокой частоты от мультивибратора, собранного на элементах D1.1, D1.2. На выходах схем совпадения образуются сигналы управления, заполненные импульсами высокой частоты мультивибратора (пакеты импульсов). Эти пакеты импульсов усиливаются каскадами, собранными на транзисторах VT1, VT2, и поступают на трансформатор TR1. Сигналы управления восстанавливаются выпрямителями V5—V8. Восстановленный сигнал управляет выходными ключами прерывателя, собранными на транзисторах VT3, VT4. Преобразование сигналов управления вызвано необходимостью полной гальванической

телем, будет тотчас напечатаны аппаратом. В служебной строке появляется сообщение режима; выход — клавиша КТ. В этом режиме автоматически будет подана команда аппарату произвести функции «возврат каретки» и «перевод строки» после печати 64 символов в строке.

Таким образом, программа универсальна и может быть применена для распечаток

листингов Фокала и Бейсика, текстов, подготовленных в редакторах EDASP и МИКРО, текстов, имеющих свою систему чтения — READER, МОНСТР, ОБЛОЖКА и т. д., для вывода машинных кодов из монитора МСТД, печати тестов.

К наиболее, на мой взгляд, интересным применениям программы в других областях относится использование программно-аппа-

1000A2000AH=013307

\$1000A2000L

005037	000054	012737	100112	000030	012737	101362	000274
010704	062704	001532	112400	001402	104016	000774	010704
062704	001400	012701	000001	112400	001403	104022	005201
000773	005002	104020	104006	010701	062701	000074	010137
000030	122700	000040	001026	022737	012701	120000	001002
000137	120234	005000	012701	120020	012120	022701	120046
002374	012737	177777	000262	010700	100404	062700	177624
010037	001744	000000	010546	010705	062705	000340	010537
000274	016605	000002	024527	104016	001445	021527	104020
001420	021527	104014	001133	013746	000030	013746	000054
012737	100112	000030	104014	012637	000054	012637	000030
000420	010046	010346	010203	000303	042702	177400	001002
052702	020000	112100	104016	120300	001401	077205	012603
012600	012605	000002	010046	042700	177400	122700	000025
001003	004767	000276	000462	105737	000054	001457	122700
000040	101011	122700	000200	101027	122700	000240	101003
122700	000300	101421	122700	000012	001003	005037	000026
000433	105737	000055	001030	122700	000011	001405	122700
000014	001024	001270	000012	013700	000156	042700	177700
023700	000026	101406	010046	012700	000012	004767	000320
012600	010037	000026	011600	004767	000304	012600	012605
000137	100112	012700	000137	000747	122737	000014	177662
001004	022737	121110	000004	001427	122737	000016	177662
001410	122737	000017	177662	001351	004767	000046	000137
101362	004437	110346	105137	000054	005000	104022	004767
000010	004437	110362	000002	000207	012700	000012	004767
000156	000207	004767	177762	005000	104022	010704	062704
000630	012701	000007	112400	001403	104022	005201	000773
005067	000532	104006	104016	122700	000003	001415	122767
000100	000512	001414	122700	000012	001416	062767	000001
000474	004767	000052	000756	005000	104022	000207	005067
000454	004767	000010	000765	005067	000442	000762	012701
000057	004767	000204	012701	000073	004767	000174	000207
004437	110346	042700	177400	122700	000012	001505	010701
062701	000234	020027	000043	002472	020027	000100	002420
020027	000200	002410	020027	000300	002461	062701	000075
042700	000340	000407	062701	000035	042700	000140	000402
162700	000043	050001	111101	042701	177400	010102	042702
177774	020267	000124	001412	010267	000116	060702	062702
000252	010146	111201	004767	000020	012601	004767	000006
004437	110362	000207	006201	006201	012702	000010	006201
106737	177714	012703	005500	077301	077207	000207	012701
000067	004767	177744	000754	012701	000057	004767	177732
012701	000073	004767	177722	000743	000002	156665	156735
102725	156555	116565	016745	046425	062505	126765	052575
146705	106475	156615	156415	156465	033346	133216	113366
057056	123316	067206	117036	047076	127106	077326	007306
013146	073126	156735	156735	122735	033747	133617	113767
057457	123717	067607	117437	047477	127507	077727	007707
013547	073527	112455	126335	004725	037401	000000	000000
050040	044522	052116	046456	045524	020062	053040	027061
027062	030440	034471	143462	020056	052270	050123	067543
070162	134056	000000	020270	142760	140736	154324	151440
145440	140714	144727	152301	151325	035331	134040	173440
144331	142317	036055	172353	005076	000000	170014	177345
172341	020370	175351	170040	167762	171347	166741	174755
160455	031362	171057	171765	005056	162760	160776	174364
171440	165440	160754	047677	172341	171365	026771	171341
027462	160754	027364	173412	164371	162357	173440	166440
167357	172351	171357	164440	164754	166440	162363	036055
173767	162357	027076	173412	163040	165757	166341	164440
164754	161040	165345	164763	026753	171360	161357	166345
005056	174767	167750	020344	160756	172040	166345	172345
165341	035360	172440	027760	030741	005066	037075	000000

ратного комплекса на предприятиях связи в качестве установки для безбумажной обработки телеграмм, с последующей передачей их на пункты коммутации сообщений (ЦКС). Комплекс обеспечит высокую стабильность скорости и качество при такой работе, несравнимое с механическим телеграфным аппаратом.

Ю. КОТОВ
(Москва)

Простое соединение ПЭВМ типа IBM PC и БК-0010

На фоне преимущественного распространения ПЭВМ типа IBM PC машины малой мощности, вроде БК-0010, «Микроши» или других 8-разрядных бытовых компьютеров, остаются значительно более дешевыми и доступными; именно с них начинается «народная компьютеризация» да и учебный процесс в школах по соответствующим предметам. Применимы малые ПЭВМ и в проектировании, на производстве для решения несложных задач — таких, для которых привлечение ПЭВМ большей мощности и стоимости было бы расточительно. Но в ситуации использования ПЭВМ разной мощности возникает естественная задача, если непереносимости программного обеспечения с одних машин на другие в полном объеме, то передачи информации и оперативного сопряжения, когда одна машина может быть главной, а другая вспомогательной — использоваться для предварительной подготовки информации, просмотра результатов одним пользователем при работе на основной ПЭВМ другого, для управления дополнительным оборудованием и т. д. В учебном процессе и даже в профессиональном (или пользовательском) программировании при личном владении малой ПЭВМ учащийся или пользователь может дома подготавливать информацию или, на совместимом языке, программы, части программ, затем пересылать их на ПЭВМ большей мощности и там доотлаживать и включать в эксплуатацию (или сдавать преподавателю).

Довольно эффективна передача информации через диски, хотя само подключение дисководов, да еще совместимого с дисководом IBM PC, довольно сложное и дорогое мероприятие. Да и не всегда «дисковый обмен» заменяет оперативную связь.

Электрически персональные компьютеры соединяют, как известно, различным обра-

Драйвер может быть также использован коротковолновиками для работы в режиме телетайпа на передачу заранее набранных сообщений или прямой работы в эфире.

Все замечания по статье и идеи о новых областях применения можно направить автору по адресу: 612220, Кировская обл., г. Яранск, а/я 2, Таланову Сергею Павловичу.

зом — даже с использованием модемов и телефонной связи. Более сложна задача сопряжения одной ПЭВМ большей мощности с несколькими малыми, что требует какой-то — программной или электронной — коммутации. Сопряжение в рамках выпускавшегося КУВТа центральной ПЭВМ ДВК-2 с малыми БК-0010 вылилось в целую плату, вставляемую в корпус ДВК-2 и по размерам такую же, как плата всей основной электроники той же ДВК-2, да еще и в 12 дополнительных схем в корпусах, подключаемых к БК0010.

В условиях же кризиса и беспредела в отношении взвинчивания цен — тем более различными кооперативами и «умельцами» — автором была поставлена задача максимально простого сопряжения БК-0010 с ПЭВМ типа IBM PC.

Соединение производится напрямую, через параллельные порты. БК-0010, как известно, имеет такой порт на 16 входных и 16 выходных разрядов. Типовая IBM PC имеет «неполный» параллельный порт, с 8 выходными информационными линиями, а также несколькими выходными и несколькими входными управляющими (и контрольными) линиями. Этот порт обычно используется для управления печатающим устройством, иногда — графопостроителем. На низовом уровне информационные и управляющие разряды работают одинаково; напряжение на линиях (2,5—5 В) соответствует напряжениям, принятым в БК-0010 (стандарт TTL). В 25-контактном разъеме параллельного порта IBM PC найдены пять контактов для входных сигналов. ПЭВМ БК-0010, питающаяся через трансформатор, электрически изолирована от «земли», так что соединение общего вывода двух машин проблем не вызывает.

Альтернативное использование последовательного интерфейса хотя и уменьшает

количество проводов для связи, но вызывает дополнительные вопросы. Для оптронной схемы согласования нужно «нестандартное» напряжение ± 16 вольт. В стандартной БК-0010 на наружный разъем не выведены сигналы последовательного порта (хотя в схеме машинки соответствующий «порт» имеется, а в программе-мониторе, зашитой в ПЗУ, есть драйвер телеграфного канала).

В соответствии с имеющимися пятью входными (со стороны IBM) линиями и для уменьшения количества соединительных проводов было принято решение передавать (в обе стороны) за один раз по полубайта — по 4 разряда. Пятый разряд в каждую сторону используется как управляющий (так называемый стробирующий); этот же разряд при передаче в обратном направлении используется как сигнал готовности (ответный). Дополнительно один провод соединяет общие контакты («землю»). Всего 11 проводов. Из покупных деталей может понадобиться разве что 25-контактный разъем для ПЭВМ IBM (не обязательно с позолоченными иглоками); разъем для подключения к БК-0010 в ее комплекте имеется. Кроме этого, можно перепаять разъем-заглушку, даваемый для тестовой проверки машины. Для экспериментов при наблюдении должной осторожности можно обойтись без разъема к IBM PC, индивидуально вставляя в дырочки залуженные концы проводов.

Согласно распределению контактов в обоих разъемах (их номера оттиснуты на той стороне разъемов, где производится пайка) соединение таково:

Номера контактов	IBM PC	БК-0010	Назначение
	1 _____	V31	«строб» от IBM PC
	2 _____	B24	данные, 0 разряд от IBM
	3 _____	A24	» 1 разряд
	4 _____	B23	данные, 2 разряд
	5 _____	B17	» 3 разряд
	15 _____	B5	«строб» от БК-0010
	13 _____	A16	данные, 0 разряд от БК
	12 _____	A13	» 1 разряд
	10 _____	B12	» 2 разряд
	11 _____	B10	» 3 разряд
	18—25 _____	A11	«общий»

«Строб» от IBM IC ведет к 8-му входному разряду БК, «строб» от БК-0010 берется от 4-го выходного разряда БК.

IBM PC могут иметь варианты вывода на разъем каналов COM1 или COM2. Обычно задействован COM1 и порты с 16-ричными адресами 378, 379, 37A. Порт 378 — регистр выходных данных параллельного интерфейса, его разряды 0—7 выведены на контакты разъема 2—9. Порт 379 — регистр статуса (входной); в прочитанном из него байте

разряды 3, 4, 5, 6, 7 соответствуют контактам 15, 13, 12, 10, 11 разъема. При выводе на печать они соответствуют сигналам «ошибка», «устройство подключено», «нет бумаги», «подтверждение приема символа», «устройство занято». Важно заметить, что 7-й разряд в порту инвертируется, т. е. принимает противоположное значение переданному. Кроме того, все сигналы в БК инвертируются, т. е. логический ноль соответствует напряжению 5В, а единица — низкому уровню. Из младших, неиспользуемых разрядов порта обычно читаются единицы. Потому после отправки нулевого кода от БК из порта прочтется 16-ричное число 7F; 16-ричное число 31 даст байт 87. Для выделения информационного полубайта (после проверки 3-го разряда) код надо сдвинуть направо на 4 разряда и произвести операцию XOR («исключающее или») с операндом 8₍₁₆₎. Вместо сдвига код можно нацело разделить на 32.

Порт с адресом 37A — регистр управления (выходной); в нем можно использовать до 5 разрядов, хотя в данном случае применяется только нулевой (стробирующий) импульс.

Для отправки информации в порты 378 и 37A на языке Бейсик используются команды OUT a, k, где a — адрес порта, k — посылаемый код. Для чтения информации из порта используется функция kl=INP(a). На Ассемблере послать информацию в порт можно, например, командой OUT DX, AL, если предварительно в регистр DX занесен адрес порта, а в AL — нужный код. Читается

информация, например, командой IN AL, DX (в DX также предварительно заслан адрес, в AL будет результат).

Со стороны БК-0010, как известно, входной и выходной порт имеет восьмеричный адрес 177714 — формально являющийся как бы адресом одной из ячеек оперативной памяти (в IBM PC адресация портов отделена от адресации оперативной памяти). На Бейсике для отправки информации в «порт» используется инструкция

POKE&O177714, kod %; для чтения — функция: kod %=PEEK (&O177714).

При организации процесса на низком уровне программы обмена информацией сами должны посылать и принимать информацию, управляющие и ответные сигналы, ждать друг друга и осуществлять взаимную синхронность работы. В то же время для программиста остается больше свободы, чем при попытках использования функций операционной системы или BIOS, либо же таких «готовых» операторов Бейсика, как LPRINT.

Передача информации по полбайта, конечно, медленнее, чем по байту; в направлении от IBM к БК организовать побайтовую передачу в принципе не сложнее, хотя надо протаскивать больше соединительных проводов.

Целые числа в ПЭВМ обоих типов кодируются одинаково и, кроме упомянутой выше перекодировки, при чтении из порта 379 другой перекодировки не требуется: надо только разделить число на полубайты и затем слить их обратно. При передаче символьной информации надо иметь в виду, что русские буквы и специальные символы псевдографики во второй половине таблицы кодов кодируются различно; в частности, в БК русские буквы — не по алфавиту. Потому для кодов, чье числовое значение больше 127, нужна перекодировка.

Ниже приводим текст простых программ на Бейсике (для ПЭВМ IBM PC выбран вариант Турбо-Бейсика) — для передачи массива чисел из БК-0010 к IBM.

1 REM ТЕСТОВАЯ ПРОГРАММА ПЕРЕСЫЛКИ МАССИВА ЧИСЕЛ ИЗ БК В IBM (В БК-0010)

2 DIM A%(100) *Задание массива для пересылки

3 DATA 1,0,-1,123.436,.....

4 FOR I%=1 TO 8

5 READ A%(I%)

6 NEXT

9 POKE &o177714,0% *Очистка "порта"

10 INPUT "Введите номер первого параметра и кол-во чисел"; A1%,A2%

11 I1%=A1%

12 C%=A%(I1%) *Выделение числа из массива

13 C1%=(C% AND &o77777)\256% *Выделение байта

14 IF C%<0% THEN C1%=C1%+128% *Добавить знак

15 C2%=C% AND &o377 *Выделение второго байта

16 ? I1%;C%;C1%;C2% *При желании вывод на экран

17 D%=C1%

18 GOSUB 100 *Передача первого байта

19 D%=C2%

20 GOSUB 100 *Передача второго байта

21 I1%=I1%+1% *Движение в массиве

22 IF I1%<=A2% GOTO 12 *К передаче следующего числа

23 POKE &o177714,1% *Сигнал конца передачи

24 ? "Конец;передано";i1%-1%;"чисел"

25 STOP *Останов или продолжение программы

100 D1%=D%\16% *Подпрограмма передачи байта; выделение полубайта

101 J%=0%

102 IF PEEK(&o177714)<256% GOTO 102 *Ожидание от IBM

103 POKE &o177714,D1%+16% *Посылка кода с упр. битом

104 IF PEEK(&o177714)>255% GOTO 104 *Ожидание

105 POKE &o177714,0% *Сигнал в IBM

```

106 JX=JX+1%
107 D1%=DX MOD 16%
108 IF JX=1% GOTO 102
109 RETURN

```

```

^Выделение другого полубайта
^К передаче второго полубайта

```

ПРОГРАММА ПЕРЕСЫЛКИ МАССИВА ЦЕЛЫХ ЧИСЕЛ ИЗ БК В ИВМ

```

dim m(100)           ^В этот массив пересылка
CLS
OUT &H37A,0         ^Обнулить "порт"
INPUT "Ожидание";t% ^Запустить прог. в БК
JX=1: JJX=0
10 JJX=0
16 OUT &H37A,1       ^Разрешение в БК
20 KX=INP(&H379): if kX=&h6f goto 80 ^Ожидание кода от БК
IF (KX AND &H8)<>0 GOTO 20
KX=INP(&H379): KX=KX XOR &H7F: K1X=KX\16 ^Прием и преобр. кода
OUT &H37A,0         ^Сигнал в БК
30 KX=INP(&H379): IF (KX AND &H8)=0 GOTO 30 ^Ожидание продвижения от БК
JJX=JJX+1: IF JJX=1 THEN K2X=K1X:GOTO 16 ^Надо ли второй полубайт?
KX=K2X*16+K1X      ^Монтаж байта
IF JJX=0 THEN KKKX=KX%: GOTO 55
C%= (kkkX and &h7f)*256+kkX ^Монтаж числа
if (kkkX and &h80)<>0 then C%=C% or &h8000 ^Если отрицательное
? JX: C%           ^По желанию - вывод на экран
M(JX)=C%: JX=JX+1 ^Занесение в массив
IF JX>100 GOTO 60  ^Если переполнение массива
55 jJX=1-jJX: GOTO 10 ^К приему следующего числа
60 ? "Переполнение массива": lX=a2%: STOP ^Обработка переполнения массива
80 ? "Конец; передано чисел"; jX-1 ^Принят код конца передачи
..... продолжение программы .....

```

Программы можно запустить в любой последовательности, и они (передающая и приемная) затем ждут друг друга и продвигаются в работе синхронно, посылая и контролируя управляющие и ответные сигналы. Кстати, для некоторых вариантов можно несколько ускорить работу программы, передавая информацию и при единичном, и при нулевом управляющем (стробирующем) сигнале. В программах же, приведенных ниже, информационный сигнал при выключенном стробе используется для передачи признака окончания работы (вся ин-

формация передана). Он же может использоваться для передачи контрольной информации (разряда четности или др.). Сходным образом устроены программы передачи от ИВМ к БК.

Более быструю передачу обеспечивают, конечно, машинные программы; для ИВМ РС их можно подготовить на Ассемблере, для БК — непосредственно в машинных кодах.

Более сложные программы (очевидно, со стороны БК — в машинных кодах) подготовлены для передачи текстов бейсиковских

программ. Известно, что при загрузке бейсиковской программы в память БК она там хранится в специальном виде; команды (операторы) заменены условными численными кодами, последовательные строки расположены в «случайном» порядке (обусловленным порядком их ввода и редактирования), а номера строк и адреса их размещения располагаются в специальной таблице — ближе к старшим адресам доступной памяти. Для вывода набора данных с последовательными строками необходима некоторая реконструкция программы, подобная той, которая совершается при действии оператора LIST. Кстати, отдельные блоки операционной системы БК-0010 можно использовать в программе реконструкции. Аналогично при приеме программы на Бейсике от IBM PC в БК надо ее ввести в память соответствующим образом, построить таблицу адресов строк и др.

Были и другие предложения, например, действовать через файлы с расширением ASC., т. е. передавать информацию как содержимого файла, записанного прежде командой SAVE, или сформировать такой файл, который затем будет читаться командой LOAD.

Отдельный вопрос о частичной совместимости Бейсиков, особенно если иметь в виду Бейсик БК-0010-01 и Турбо-Бейсик. Турбо-

Бейсик позволяет (как в БК) нумеровать все строки (но не сортирует их по номерам) и большинство операторов БК обрабатывает правильно. Есть разница, например, в последовательности параметров для операторов LOCATE (да и рабочее число строк и столбцов другое), но это отдельные детали. Список подобных отличий и ограничений можно продолжить. Более постоянная особенность — различия в обработке синтаксиса. БК как бы игнорирует пробелы в записи инструкций и программисты пользуются их устранением для экономии памяти, сливая операторы (команды) с именами переменных и т. д. Турбо-Бейсик же такой «каши» не допускает. С другой стороны, в БК-0010 оказываются недопустимыми такие переменные, как IN или RU.

Некоторые отличия можно подправить или компенсировать специальной программой типа «претранслятора» или «конвертера».

Работа показала, что при соблюдении специальной инструкции несложные программы, работающие на БК, оказываются обрабатываемыми и Турбо-Бейсиком.

Связь разных ЭВМ на программном уровне позволяет не ограничиваться задачами пересылки данных или программ, но и создавать пары совместно работающих программ с различным разделением задач между машинами.

ВНИМАНИЮ ПОЛЬЗОВАТЕЛЕЙ УКНЦ (МС0202), АГАТ 7/9 (НГМД 532301)

Научно-производственный центр «Тема» предлагает для школ, лицеев, гимназий, педагогических училищ и ПТУ следующие программные продукты:

Тесты для детей 7—9 лет

Тесты для профессиональной ориентации (8—11 кл.)

Специализированные тесты для диагностики умственного развития (7—11 кл.)

Обучающие и контролирующие курсы:

Математика (2—3 кл.)

Русский язык (2—3 кл.)

Алгебра (9—11 кл.)

Геометрия (9—11 кл.)

Системные программы

Программное обеспечение можно приобрести как наложенным платежом (безналичная форма расчета), так и по перечислению.

По заявке бесплатно высылаются каталог программных средств и наложенным платежом демонстрационная дискета

Заказы принимаются по адресу:

Россия, 346407, г. Новочеркасск, Ростовская обл., ул. Ленгника, д. 9.

НЦП «Тема», отдел информатики.

Тел.: 863-52-47209 (45349, 46225)

Украина, г. Горловка, Донецкой обл.

Тел.: 062-42-20081

Не забудьте в заявке указать тип ЭВМ и дисковод!

Разрабатываем программы по вашему заказу.

Ждем ваших заявок и предложений.

Фирма «Колледж»:

**„Мы помогаем Вам в решении
проблем компьютеризации
учебных заведений”**

К КОЛЛЕДЖ

К вашим услугам:

**Поставка компьютерных классов на базе
БК-0011М, УКНЦ, IBM-совместимых
компьютеров**

**Розничная продажа комплектующих для
БК-0011М, УКНЦ.**

**Комплексное программно-техническое обслужи-
вание СВТ любых типов (для Москвы и Москов-
ской области)**

**Изготовление мебели для учебных заведений
(парты, столы, стулья, скамейки, стеллажи
и др.)**

**Наш
адрес:
107005,
Москва
Волховский
пер., 11
Фирма «Колледж»
телефон: 267-7058**

**Приглашаем организации и частных лиц из
различных регионов СНГ к сотрудничеству в
области компьютеризации учебного процесса:**

- ◆ создание филиалов по обслуживанию СВТ
в Вашем регионе
- ◆ совместная разработка программного обес-
печения по различным учебным дисципли-
нам

МЫ ПОСТАВИМ ВСЮ НЕОБХОДИМУЮ ТЕХНИКУ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, И СДАДИМ СИСТЕМУ "ПОД КЛЮЧ"!

Фирма "ЛинТех" представляет новейшие сетевые системы на основе стандартных КУВТ "КОРВЕТ", "НЕЙВА", "ФОРМАНТА", "ОРБИТА" и IBM-совместимых компьютеров для школ, техникумов, вузов и промышленных предприятий.

ВнастоящиймоментсуществуюттримодификациисетевойсистемыNET-CP/M:

ОС NET-CP/M B-1.0 - многопользовательская операционная система с головной машиной типаIBMPC/XT/AT/286/386/486итерминалами"КОРВЕТ"(доб4мест),снабженнымивысокоскоростными сетевыми адаптерами (375 Кбод). В настоящее время разработаны мощная обучающая система, сетевая СУБД, табличный и текстовый редакторы, электронная почта, подготавливаются учебные курсы по ряду предметов. Система открыта для развития приложений на базе единого графического интерфейса. Возможность работы со стандартными программами CP/M сохранена в файловой системе MS-DOS IBM PC. Предназначена для использования в качестве общеучрежденческой (в том числе и на крупных промышленных предприятиях) локальной сети, совмещающей обработку деловой информации и процесс обучения, представляет собой качественно новую систему, соответствующую современному уровню развития сетевых технологий.

ОС NET-CP/M A-1.0 - сетевая версия ОС CP/M для КУВТ стандартной конфигурации. Позволяет на РМУ работать в ОС CP/M, обращаясь к дискам РМП. Все возможности стандартной CP/M сохранены. Отличительные особенности - мощная система разграничения доступа, высокая скорость обмена по сети (при стандартном сетевом оборудовании), стопроцентная защита от сбоев сети и дисков, высокие сервисные возможности. Предназначена для ведения учебного процесса.

ОС NET-CP/M AB-1.0 - промежуточный этап между A-1.0 и B-1.0 представляет собой NET-CP/M A-1.0 с установленными высокоскоростными (375 Кбод) сетевыми адаптерами. Эта система позволяет с минимальными затратами заменить в последствии головную машину на IBM-совместимую и получить NET-CP/M B-1.0

Наряду с сетевыми системами фирма "ЛинТех" поставляет разнообразные программные продукты для выше названных КУВТ, отличающиеся высочайшим качеством исполнения, обилием графики, высокой надежностью.

Ассоциация "ЛинТех" продолжает регистрацию пользователей упомянутых КУВТ, объявленную в "ИНФО" #1 '92. Ассоциация ставит своей целью объединение пользователей для максимального удовлетворения их потребностей в программном и аппаратном обеспечении, создание условий для обмена информацией. Для членов ассоциации будет функционировать "горячая линия" - по телефону мы сможем ответить на вопросы по программному и аппаратному обеспечению нашей фирмы.

Кроме того, члены ассоциации могут получить бесплатный пакет игровых программ при условии оплаты дискет и почтовых расходов.

Для регистрации достаточно вырезать, заполнить и выслать в наш адрес анкету.



Адрес для корреспонденции: 119501, Москва, а/я 942

Телефон для справок: (095) 273-50-14

Телефакс: (095) 292-65-11 LINTENBox9318

Представитель в республике Беларусь МП "Корвет"

231100 Гродненская обл., г. Ошмяны, ул. Красноармейская, 4-5, т. 2-15-03

линия отреза -----

РЕГИСТРАЦИОННАЯ АНКЕТА УЧАСТНИКА АССОЦИАЦИИ "ЛИНТЕХ".

Адрес: _____

Телефон: _____ Телефонфакс: _____

Название организации: _____

ФИО ответственного лица: _____

Используемая вычислительная техника: _____

Используемое программное обеспечение: _____

Регистрационная анкета ОБЯЗАТЕЛЬНО должна быть заверена круглой печатью Вашей организации и отправлена по адресу: 119501, г. Москва, а/я 942.

С. КОНОВ

«ЛИнТех» представляет: Локальная сеть с IBM-совместной головной машиной NET-CP/M версия В-1.0.

В первом номере журнала «ИНФО» за 1993 г. в статье «Сетевая операционная система «NET-CP/M» версия А-1.0 была сделана попытка ознакомить пользователей КУВТ «Корвет» («Форманта», «Нейва», «Орбита») с сетевой операционной системой (ОС). Из существующих на настоящий момент для данного класса компьютеров — ОС NET-CP/M А-1.0 — это одна из лучших разработок.

Основная цель создания ОС NET-CP/M А-1.0 — программно компенсировать отсутствие дисководов на рабочих местах учеников (далее будет применяться термин «терминал») посредством использования дисководов рабочего места преподавателя (далее будет использоваться термин «головная машина» — ГМ). При этом на терминале обеспечивалась работа в среде ОС, совместимой со стандартной ОС CP/M-80.

ОС NET-CP/M А-1.0 представляет собой программный комплекс, работающий без каких-либо переделок на стандартном оборудовании КУВТ. Несомненно, КУВТ «Корвет» является одним из самых надежных школьных классов на настоящий момент (кто не верит, пусть попробует сравнить его, скажем, с УКНЦ), компьютеры имеют мощные графические возможности, быстрый алфавитно-цифровой экран, гибкую архитектуру, под них написано достаточно большое количество программ.

При всем при этом есть три крупных недостатка КУВТ, отражающихся на работе сети в целом и не позволяющих создать действительно функционально мощную сетевую ОС. Прежде всего низкая физическая скорость обмена (19.5 кбит/с) не дает даже одному пользователю в сети производить пересылки относительно больших (к примеру, программа интенсивно использует оверлеи) объемов информации без риска заснуть око-

ло терминала. Ну, а для нескольких пользователей сети получится великолепный сюжет для кошмарного сна преподавателя информатики — к концу урока все ученики, наконец, загрузили свои машины. Если же все-таки удалось пропихнуть свою информацию в сеть, то там ждет вторая неприятность в лице низкопроизводительной головной машины — не торопясь она считывает данные из сети, немножко отдохнет и, наконец, обратится за помощью к дисководам. Ну а все, что связано с дисководами, — это отдельная и очень грустная песня: если класс снабжен японскими дисководами, то в этом случае навредить может крайне ненадежный контроллер дисковода, ну, а если отечественное происхождение имеют и дисководы, то по отношению к пользователям более гуманно выкинуть весь КУВТ, нежели пытаться его использовать.

Резюме для вышесказанного — используя возможности стандартного оборудования КУВТ, обеспечить чисто программными средствами высокое быстродействие, надежность и функциональную полноту не представляется возможным. Необходим переход на качественно более высокий уровень в аппаратном, и, как следствие, в программном обеспечении сетевых систем в этот переход был сделан...

В конце 1992 г. Лабораторией Информационных Технологий («ЛИнТех») закончена разработка программно-аппаратного комплекса, представляющего собой качественно новый уровень развития сетевых систем. Были ликвидированы узкие места аппаратуры КУВТ, написана новая, функционально мощная сетевая операционная система NET-CP/M В-1.0, столь же похожая на ОС NET-CP/M А-1.0, как и калькулятор на IBM-PC.

Здесь уместно будет сказать, что ОС NET-CP/M В-1.0 разрабатывалась для применения в качестве дешевой общеуниверсальной локальной сети, что обуславливает изначально заложенные высокие технические требования к системе. В настоящее время готовятся программы автоматизации бухгалтерского и складского учета, учета кадров, информационно-справочная система, мощная обучающая система.

В качестве головной машины в ОС NET-CP/M В-1.0 используется IBM-PC/AT совместимая машина. В головную машину вставляется сетевой контроллер. К каждому терминалу через порт расширения подключается сетевой контроллер терминала. Всего в сети к одной головной машине IBM-PC может подключаться до 64 терминалов, а длина магистрального кабеля может составлять 1 км.

Сетевые контроллеры предназначены для повышения физической скорости передачи данных в канале и обеспечивают скорость 375 кбит/с, что почти в 20 раз выше скорости стандартной сети КУВТ. Для установки сети в школьном классе не требуется перемонтажа существующих линий связи — новая сеть работает на коммуникациях старой, что позволяет в случае необходимости (например, неисправность IBM-совместимой головной машины или сетевых контроллеров) вернуться в режим работы с ОС NET-CP/M А-1.0.

Итак, три основные проблемы аппаратуры КУВТ решены — высокая скорость обмена в сети обеспечивается сетевыми контроллерами, проблема производительности головной машины и надежности дисководов отпадает по определению — IBM-совместимые машины высокопроизводительны, винчестеры надежны, имеют большую емкость и скорость доступа к данным.

Тем не менее изменения в аппаратном обеспечении сети на самом деле отображают только вершину айсберга — на высокопроизводительном оборудовании можно разрабатывать и качественно более мощные программные системы. Исходя из этого принципа и создавалась ОС NET-CP/M В-1.0, которая представляет собой сетевую операционную систему с иерархической файловой структурой и системой резидентных команд практически полностью аналогичных операционной системе MS-DOS IBM-совместимых компьютеров. При этом сохранены все возможности и полная программа совместимости с ОС CP/M, что позволяет использовать все существующие для ОС CP/M программное обеспечение, но только существенно более эффективно.

Рамки журнальной статьи не позволяют подробно описать все возможности систе-

мы, поэтому для понимания используемых в ОС NET-CP/M В-1.0 базисных понятий ОС MS-DOS (например, понятие иерархической структуры каталогов) необходимо ознакомиться с соответствующей литературой по ОС MS-DOS, которой в настоящее время превеликое множество.

Итак, программное обеспечение (ПО) сети состоит из ПО головной машины и ПО терминала. В свою очередь ПО головной машины состоит из управляющей программы (монитор) головной машины и набора сервисных утилит по обслуживанию сети. ПО терминала состоит из управляющей программы (монитор) терминала, который в конечном итоге для пользователя и является той самой ОС, с которой он работает на терминале.

ПО головной машины подробно описано в фирменном руководстве по ОС и представляет интерес скорее для администратора сети, а не для пользователя, поэтому имеет смысл заострить внимание только на одном моменте — на понятии резидентности монитора ГМ. Монитор ГМ представляет собой основную программу, управляющую всеми режимами работы сети, исполняющую сетевые запросы с терминалов, обрабатывающую ошибки, и термин резидентности монитора ГМ означает, что на ней параллельно может работать и монитор ГМ в фоновом, незаметном для пользователя ГМ режиме и любая другая программа IBM-PC (например, игровая).

Основной интерес представляет характер работы пользователя на терминале, возможности, предоставляемые монитором терминала, количественные показатели работы сети. Это лучше всего проследить для обеих систем в сравнении с самого начального момента — с включения питания терминала.

При работе в ОС NET-CP/M А-1.0 после включения питания автоматически производится загрузка монитора на терминал. При этом разграничение ресурсов производится по принципу сопоставления физического номера терминала и USER-подобласти рабочего диска. Естественно, что если несколько пользователей последовательно работают на одном и том же терминале и рабочая дискета не меняется, то все файловые операции производятся в одной и той же USER-подобласти диска. В современных сетевых системах такой принцип разграничения доступа недопустим.

В ОС NET-CP/M В-1.0 к основному принципу работы системы разграничения доступа относится соответствие имени и пароля пользователя, с одной стороны, и ресурсов головной машины — с другой. При этом сразу после включения питания терминала появляется запрос на ввод имени и пароля. Пос-

ле этого на ГМ производится проверка имени и пароля, и только в случае их корректности возможна регистрация терминала в системе.

Регистрация терминала в системе состоит из трех фаз — определение доступных для данного пользователя дисков, пересылка резидентной части монитора терминала на терминал и исполнение на терминале стартовой последовательности команд.

Остановимся подробнее на каждой из трех фаз регистрации терминала в системе.

Всего на терминале пользователю доступно четыре диска <A, B, C, D>. Диск <A> соответствует электронному диску терминала, и он всегда доступен как на чтение, так и на запись. Три остальных диска составляют с подкаталогами винчестера или флоппи-диска головной машины, определяемыми администратором сети в специальном файле регистрационной информации пользователей и могут быть защищены на запись или модификацию. Эти подкаталоги являются для пользователя терминала корневыми для данного диска.

Пересылка резидентной части монитора терминала на терминал, с точки зрения пользователя, выглядит просто как банальная загрузка ОС и пояснения не требует.

Третья фаза загрузки — исполнение стартовой последовательности команд состоит в запуске на терминале командного файла (аналогично запуску AUTOEXEC.BAT в системе MS-DOS) и исполнение заложенной в нем последовательности команд. При этом командный файл для каждого пользователя может быть своим и определяется администратором сети в файле регистрационной информации пользователей.

Поясним все вышесказанное на примере регистрации в системе трех пользователей — создателей системы NET-CP/M В-1.0 — Кононова С. В., Нужи В. В и Авдюшина С. С.

Прежде всего всем им нужно будет зарегистрироваться у администратора сети. Для этого каждому необходимо придумать имя (можно не придумывать, а взять настоящее) и пароль, определить желаемое соответствие между именами дисков и подкаталогами винчестера, создавать стартовый файл и сообщить его имя.

Пусть структура каталога винчестера головной машины соответствует приведенной на рисунке.

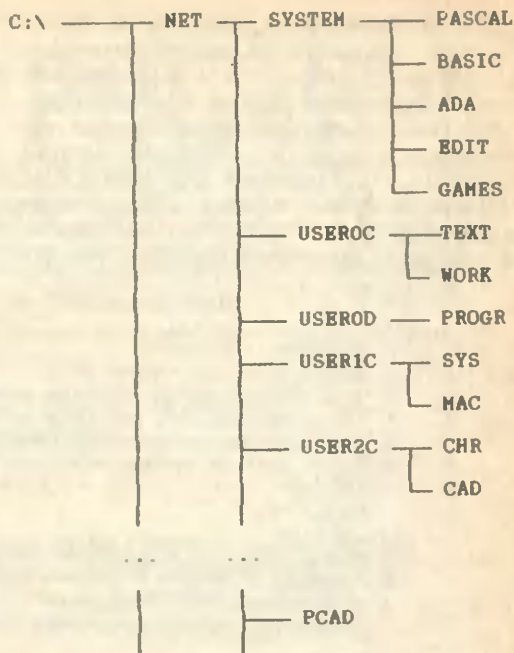
В этом случае данные в файле регистрационной информации пользователей могут быть следующими:

для г-на Кононова

Имя: Вождь

Пароль: ЛИНТЕХ

Стартовый файл: AUTO1.BAT



Диск — C:\NET\SYSTEM (нет доступа на запись)

Диск <C> — C:\NET\USEROC (есть доступ на запись)

Диск <D> — C:\NET\USEROD (есть доступ на запись);

для г-на Нужи

Имя: Вася

Пароль: А

Стартовый файл: AUTO2.BAT

Диск — C:\NET\SYSTEM (нет доступа на запись)

Диск <C> — C:\NET\USERIC (есть доступ на запись)

Диск <D> — не определен (полностью заблокирован);

для г-на Авдюшина

Имя: г-н Авдюшин

Пароль: Сергей Сергеевич

Стартовый файл: AUTO3.BAT

Диск — C:\NET\SYSTEM (нет доступа на запись)

Диск <C> — C:\NET\USER2C (есть доступ на запись)

Диск <D> — не определен (полностью заблокирован).

Итак, г-н Кононов может с любого терминала, правильно введя имя и пароль и автоматически запустив при этом файл AUTO1.BAT, получить доступ к трем директориям винчестера головной машины и, естественно, ко всем подкаталогам и файлам, которые в нем находятся. При этом соответствие между диском и кодкаталогом C:\NET\SYSTEM для всех троих опре-

делено одинаково, диск защищен на запись и может содержать общеупотребительные программы, а диски <C, D> являются личными недоступными другим пользователям.

Как уже говорилось выше, система резидентных команд и механизм запуска командных (BAT)-файлов ОС NET-CP/M В-1.0 практически полностью соответствует стандарту ОС MS — DOS. Рамки журнальной статьи не позволяют подробно описать

DATA — текущая дата,
TIME — текущее время,
VER — версия системы.

В заключение уместно привести количественные показатели работы сети. При проведении тестов использовалась IBM-совместимая машина с тактовой частотой 16 МГц, винчестером 40 МБ/18 мс/6 мс, без сопроцессора, опрашивались 32 терминала, 1 был зарегистрирован.

Выполняемое действие

Время исполнения в сети (с)

1. Регистрация в системе	0.5
2. Запуск редактора «Микро-Мир»	2.9
3. Запуск графического редактора «Акварель»	2.7
4. Запуск графического редактора «Абрис»	2.2
5. Запуск системы программирования «Express Pascal»	2.9
6. Запуск системы программирования «Си»:	
C80.COM	3.0
M80.COM	1.8
L80.COM	0.9
7. Запуск дискового варианта интерпретатора BASIC	2.1
8. Удаление 20 файлов в текущей директории	3.0
9. Копирование 200 Кбайт данных в режиме Хост-Хост через терминал в разные подкаталоги	33.0

114

каждую из этих команд, поэтому желающим вновь рекомендую обратиться к литературе по MS — DOS, а здесь приведу только перечень команд:

DIR — просмотр каталога,
TYPE — просмотр файла,
DEL — удаление файла,
REN — переименование файла,
COPY — копирование файлов,
PATH — определение путей поиска командных файлов,
CD — сменить директорий,
MD — создать директорий,
RD — удалить директорий,
CLS — очистка экрана,
FORMAT — формирование электронного диска,
ECHO — эхо-эффект при выполнении BAT-файлов,
PAUSE — пауза при выполнении BAT-файлов,
BREAK — прерывание при выполнении BAT-файлов,
REM — примечание при выполнении BAT-файлов,

В таблице, в основном, приведены сроки исполнения наиболее емких по времени сетевых запросов. Простые запросы, например загрузка небольших программ (большинство игровых, почти все системные, сервисные), исполняются практически мгновенно в пределах 0.3—0.7 с.

Приобрести систему можно у фирмы «ЛинТех», реклама которой постоянно появляется на страницах журнала «ИНФО», или у региональных дилеров, информацию о которых можно получить в вышеназванной фирме.

Р. S. Исключительно любителям использовать ворованные копии программ посвящается... Система NET-CP/M А-1.0 защищена от копирования. При несанкционированном копировании и использовании подобных копий программ никаких грозных предупреждающих сообщений выдаваться не будет, правда, при этом и система работать нормально не будет — постоянные сообщения об ошибках аппаратуры, зависания программ и т. п. — все это в наказание за грабеж от фирмы «ЛинТех».

Уважаемые читатели!

Напоминаем вам, что продолжается подписка на журнал «Информатика и образование» на второе полугодие. (N 5, 6). Индекс журнала в каталоге ЦРПА «Роспечать» для индивидуальных подписчиков 70423, для учреждений и организаций 73176 (по безналичному расчету).



НАЧИНАЙТЕ СРАЗУ С ЛУЧШЕГО!

- Оборудование самого высокого класса, выполненное по передовым технологиям;
- надежность и высокое качество комплектующих фирм INTEL, NEC, PHILIPS - мировых лидеров в области микроэлектроники;
- полная взаимозаменяемость узлов и компонентов;
- сервисная поддержка на аппаратном и программном уровнях;
- соответствующие принятым международным стандартам

ВСЕМИ ЭТИМИ КАЧЕСТВАМИ ОБЛАДАЕТ
компьютерный класс

«РУБИКОН 3.0».

выполненный на базе IBM-совместимых компьютеров с центральным процессором INTEL 80286 - 20 в архитектуре локальной вычислительной сети.

**ОВЛАДЕНИЕ СОВРЕМЕННЫМ КОМПЬЮТЕРОМ ДОЛЖНО
ПЕРЕСТАТЬ БЫТЬ РОСКОШЬЮ ДЛЯ НАШИХ ДЕТЕЙ!**

Учебный класс «РУБИКОН 3.0» - прекрасная возможность для школьников уже сегодня соединить теорию и практику, а завтра стать профессионалами международного уровня.

УЧЕБНЫЙ КОМПЬЮТЕРНЫЙ КЛАСС «РУБИКОН 3.0» - ЭТО:

1. Рабочее место учителя включает в себя высокопроизводительный персональный компьютер, обслуживающий запросы рабочих мест (рабочих станций) учеников и контролирующий производимые на этих станциях работы.
2. Рабочее место ученика предназначено для выполнения прикладных программ. Каждый учащийся получает возможность пользоваться полноценным компьютером высокого класса.
3. Современная операционная система - комплекс программ, работающих на процессор-файл-сервере (рабочее место учителя) и управляющих доступом к данным, находящимся на винчестере файл-сервера.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ СЕТИ:

- стандарт локальной вычислительной сети (ЛВС) EtherNet;
- скорость передачи 10 Мбит/с;
- топология - шина;
- кабель коаксиальный RG-58;
- максимальная длина кабеля 900 м (тонкий EtherNet);
- максимальное число станций 150 шт.

ПРЕИМУЩЕСТВА ДАННОГО УЧЕБНОГО КОМПЛЕКСА

Комплекс позволит работать и проводить обучение на программном обеспечении под управлением Windows в графическом режиме (Word Perfekt, Exel, Lotus 1-2-3, Microsoft Works и т.д.). Быстрая прорисовка графических изображений и интенсивный обмен данными позволит учащимся познакомиться и овладеть разнообразными графическими редакторами и издательскими системами последнего образца.

Фирма «РУБИКОН» осуществляет монтаж, инсталлирование, ввод в эксплуатацию и первоначальное обучение.

ГАРАНТИЙНОЕ ОБСЛУЖИВАНИЕ 12 МЕСЯЦЕВ.

НАШИ ПРЕДСТАВИТЕЛЬСТВА:

	Санкт-Петербург	Москва	Пермь	Челябинск
Адрес:	Лесной пр., д.19.	Живописная ул., д.56.	ул.Героев Хасана, д.10.	ул.Воровского, 79.
Телефон:	(812) 542-00-65, (812) 248-83-06	(095) 193-95-33	(3422) 49-26-15	(3512) 61-11-98
Телефакс:	(812) 542-21-45, (812) 542-09-89	(095) 190-63-75	(3422) 49-26-13	(3512) 61-11-98
Адрес в сети Relcom:	den@rubicon.spb.su.	andy@rubicon. exlink.msk.su.	mama@rubicon. perm.su.	

Связь по магнитофону

В кабинете информатики СШ-3 г. Чкаловска, оборудованном КУВТ «Корвет», осуществлена связь между всеми компьютерами (РМУ) по магнитофонному каналу. Данная связь дополняет штатную локальную сеть, расширяет ее возможности, может использоваться в качестве резервного способа пересылки программ на РМУ при выходе из строя учительской ЭВМ. Данная сеть обеспечивает режим «электронной почты» в пределах класса.

Расположение компьютеров в нашем классе стандартное: три ряда по четыре компьютера. Линия связи проложена экранированным кабелем. В качестве соединительных разъемов используются стандартные пятиштырьковые магнитофонные штеккера и гнезда.

В комплект компьютерного класса входит три накопителя на магнитной ленте (можно использовать любые другие магнитофоны), поэтому каждый ряд имеет свой накопитель (магнитофон).

Переключатели S1-S12 установлены на каждом РМУ. Выключатели S13-S15 установлены на столе учителя. При необходимости их можно установить в любом другом удобном месте. Если выключатели S13-S15 замкнуты, то все компьютеры соединены между собой, при размыкании осуществляется перевод соответствующего ряда в автономный режим работы.

Обеспечиваются следующие режимы работы:

1. Загрузка программы на РМУ с магнитофона. Любой из трех магнитофонов воспроизводит программу. Те РМУ, которые хотят принять эту программу, устанавливают соответствующий переключатель S1-S12 в

положение «прием», набирают и вводят команду LOAD"CAS: (можно здесь указать конкретное имя программы, можно и не указывать).

2. Запись программы с РМУ на магнитофон. Магнитофон включается на запись с клавиатуры вводится команда SAVE"CAS имя программы. Естественно, что в этот момент по линии не должно передаваться никаких других сообщений. Достоинством является то, что магнитофоны все время стоят на своих местах, нет необходимости подключать их к разным РМУ.

3. Режим «электронной почты». Компьютеры могут обмениваться программами между собой. Та ЭВМ, которая передает информацию, выполняет следующие действия: переводит переключатель S1-S12 в положение «передача», на клавиатуре набирается команда SAVE"CAS: (можно дать название пересылаемой программе, можно не указывать). Те РМУ, которые хотят получить данную программу, устанавливают переключатель в положение «прием» и вводят команду LOAD"CAS:

4. Работа ряда в автономном режиме. Если выключить соответствующий выключатель S13-S15, то все вышеописанные операции осуществляются в пределах одного ряда, каждый ряд работает независимо от других со своим магнитофоном.

Переключатели S1-S12 при работе класса лучше держать в положении «прием». Еще лучше использовать переключатели трехпозиционные, среднее положение — нейтральное.

Данная сеть работает в кабинете в течение 1,5 года. Никаких помех при работе в других режимах не вносит.

Предприятие

ТЕХНОКОМ

предлагает для УКНЦ программное обеспечение
и дополнительные технические средства.

Адрес: 103489, Москва, а/я 51.
Тел.: 536-50-51
536-50-90

А. НОВИКОВ

Такой же, только без крыльев

8. Захват

Чаще всего применяется один из трех методов «захвата»:

1. Изменение параметра.
2. Добавление фрагмента.
3. Подмена фрагмента

Наиболее простой — первый метод. Он основан на подмене в памяти одного или нескольких байтов. Для этого метода характерно сохранение длины файла и отсутствие «довесков» к программе. Однако и возможности его невелики — возможно лишь количественное изменение параметров процесса, его качественная сторона не меняется. (В крайнем случае, часть команд может быть ликвидирована заменой на команды NOP (\$EA) или изменено используемое устройство (\$C030 → \$C020)).

Вот, например, подпрограмма BELL в CM Бейсика, издающая сигнал «БИП» (здесь и далее программы написаны для Бейсик — HELLO к «Агат-7» с семибитным представлением текста):

```

FCB4-  A9 00          LDA  #$40 ; начальная
FCB6-  20 2B FB      JSR  $FB2B ; задержка
FCB9-  A0 C0          LDY  #$C0 ; количество проходов
FCBB-  A9 0C          +->LDA #$0C ; Задержка на период
FCBD-  20 2B FB      ! JSR  $FB2B ; колебаний динамика
FCC0-  AD 30 C0      ! LDA  $C030 ; звук
FCC3-  88            ! DEY  ; цикл до
FCC4-  D0 F5          +--BNE $FCBB ; исчерпания Y
FCC6-  60            RTS    ; выход из п/п
    
```

А вот программа, изменяющая тон и длительность издаваемого звука:

```

10 INPUT "ТОН (0-255)";TN
20 INPUT "ДЛИТЕЛЬНОСТЬ (0
-255)";DL
    
```

```

30 *$1800:
40 ! STA $C200
50 ! LDA # TN
60 ! STA $FCBC
70 ! LDA # DL
80 ! STA $FCBA
90 ! STA $C220
100 ! RTS
110 !:
120 CALL $1800
    
```

Команда в строке 40 открывает банк ПсевдоПЗУ по записи, команды в строках 50—80 изменяют содержимое двух ячеек CM, а в 90 банк ПсевдоПЗУ снова закрывается. Именно «банковские операции» мешают этой программе воспользоваться оператором POKE и обойтись без Макроассемблера. Если вы захотите проверить свои силы в этом методе — попробуйте заставить курсор Бейсика мигать в 2 раза чаще. Ответ вы найдете в конце статьи.

Метод добавления фрагмента предоставляет пользователю более богатый спектр возможностей, хотя это и создает дополнитель-

ные неудобства. Принцип подключения таков: вместо одной или нескольких команд изменяемой программы вставляется ссылка на блок пользователя, а закончив свою работу, этот блок «отдает долг» (выполняет те команды, вместо которых произведено подключение). Рассмотрим этот метод подробнее.

Если в месте программы, выбранном для подключения, есть трехбайтная команда, то проблем не возникает. Например, команду

```
3E28- AD 00 C0 LDA $C000
```

легко преобразовать в команды

```
3E28- 20 00 80 JSR $8000
8000- AD 00 C0 LDA $C000
8003- 30 02 BMI $8007
8005- A9 00 LDA #$00
8007- 60 RTS
```

и оттранслировать блок пользователя с любого адреса памяти (в данном случае, с \$8000. Если программа полностью расположена в ОЗУ, то новый блок лучше всего пристыковать к «хвосту» программы.). Приведенный пример полезен при запуске на «Агат-9» программ, написанных для «Агат-7». Как видите, «выполнение долга» здесь происходит в начале блока, так как здесь оно нужно самому встраиваемому драйверу. Перехват производится командами IMP или ISR. ISR позволяет вернуться к прерванной программе через команду RTS, но требует особой внимательности при обращении со стеком. IMP относится к этому более спокойно, но требует выхода из вставленного блока через IMP, а это на 2 байта длиннее, чем через RTS. Зато если трехбайтовой командой, через которую произведено подключение, тоже была IMP, то выполнение «долга» и возврат в основную программу выполняются единой IMP.

Если же в нужном месте программы нет трехбайтной последовательности, то произвести подключение будет несколько сложнее. Обычно команду IMP или ISR вставляют вместо команд длиной 2+1 или 1+2 байта;

```
65E2- A2 80 LDX #$80
65E4- 88 DEY
65E5- A9 00 LDA #$00
```

преобразуются в

```
65E2- 20 00 80 JSR $8000
65E5- A9 00 LDA #$00
8000- 38 SEC
8123- A2 80 LDX #$80
8125- 88 DEY
8128- 60 RTS
```

Если же в нужном месте программы идут только двухбайтные команды, можно вместо трех байтов записать ссылку на встраиваемый

блок, а четвертый байт заменить на \$EA, т. е. NOP (в случае, если пользуетесь ISR).

Однако далеко не к любому месту программы можно подключиться. Рассмотрим основные ограничения:

1) Если ссылка на блок вставлена вместо двух команд, убедитесь, что на вторую из них в исходной программе не было ссылок.

2) Проверьте, не попало ли обращение к новому блоку в тело нежелательного цикла, так как это может привести не только к многократным ненужным вызовам блока, но и к существенному замедлению работы программы.

3) Среди команд, взятых в «долг», нежелательно присутствие команд условной передачи управления, так как из своего нового положения в памяти они, скорее всего, до адреса перехода «не дотянутся», а использование в комплекте с ними IMP займет несколько лишних байтов памяти.

4) Еще раз напоминаю об осторожности при использовании, особенно если «в долг» берется команда, воздействующая на стек.

5) И уж совсем неприятным будет вмешательство в блоки программ, работающие в реальном времени. Если изменения в подпрограмме генерации звука приведут только к искажению издаваемого тона, то «доработка» драйверов магнитофона или НГМД может привести не только к неверному чтению данных с них, но и к непоправимой потере ранее записанных ценных данных.

Особо следует поговорить о сохранении содержимого регистров. В общем случае рекомендую сохранять в стеке или в ячейках памяти все регистры, которые в ходе работы вставленного блока могут измениться, и восстанавливать их значения перед выходом из блока (кроме тех, разумеется, ради изменения которых был написан этот блок). Однако в этом не всегда есть необходимость. Например, если вы изменяете содержимое регистра X и сразу после адреса подключения блока стоит команда LDX, то понятно, что сохранять и восстанавливать содержимое X не нужно. Еще менее «капризным» является регистр P. Если сразу после выхода из нового блока нет команды условного перехода, то P, как правило, можно не сохранять, так как большинство флагов получают свои значения непосредственно перед ветвлением (однако флаг C может при этом «сыграть с вами плохую шутку» — помните об этом).

Метод подмены фрагмента принципиально похож на только что рассмотренный метод его добавления. Разница состоит лишь в том, что после выполнения своей функции новый блок не «исполняет долги» и не воз-

вращает управление той точке программы, с которой он был вызван. Понятно, что такой перехват управления удобнее производить через IMP. (Впрочем, если новый блок не длиннее того, вместо которого он вставляется, то вполне разумно будет на адреса памяти старого блока вписать новый (или какую-нибудь его часть. В этом случае в перехвате управления вообще не будет необходимости.)

В заключение приведу пример программы, использующей данный метод. Эта программа встраивает в Бейсик «горячие клавиши», и после ее запуска многие часто набираемые команды интерпретатора вызываются простым нажатием одной из функциональных клавиш (список команд приведет в буфере BUFF как последовательность их КОИ, разделенных \$00).

```

10 * $1800:
20 ! STA $C200
30 ! LDA #$1F
40 ! STA $FD2B
50 ! STA $C220
60 ! RTS
70 ! :
80 CALL $1800

```

А для тех, кто заинтересовался предложенным ранее заданием, сообщаю ответ: в ячейку \$ED2B надо занести значение \$1F1F, и сделать это можно так:

```

100 SVX = 0: SVY = 1
110 PBF = 2: PSB = 3
1000 * $8000:
1010 ! LDA#0
! STA PSB
1020 ! STA$C200
! LDA#$4C
! STA$FD0D
1030 ! LDA# > BEG
! STA$FD0E
! LDA# < BEG
! STA$FD0F

```

```

1040 ! STA$C220
! RTS
2000 ! BEG:STY SVY
! LDY PSB
! RD:LDA BUFF,Y
2010 ! BNE EST
! LDA$19
! JSR$F85E
! JSR$FD12
2020 ! CMP#$A0
! BPL RT
2030 ! STX SVX
! STA PBF
! LDX#$7F
! LDY#0
2040 ! C1:LDA BUFF,Y
! BNE NX
! INX
! CPX PBF
2050 ! BEQ KN
! NX:INY
! BNE C1
! KN:LDX SVX
2060 ! INY
! BNE RD
! EST:INY
! STY PSB
! LDY SVY
! RT:RTS
3990 ! BUFF:$008000C5D8
4000 ! $C5C300C3
4010 ! $C1D4C1CCCCFC78D00
4020 ! $8300840085008600
4030 ! $8700880089008A00
4040 ! $8B008C008D008E00
4050 ! $8F00CCC9D3D400CC
4060 ! $CFC1C400D3C1D6C5
4070 ! $00D2D54E00D2D54E
4080 ! $8D00950096009700
4090 ! $980099009A009B00
4100 ! $D4C5D8D4BD008D00
4110 ! $8E00AAA400000000
4990 ! :
5000 CALL $8000

```

Уважаемые читатели!

Напоминаем вам, что продолжается подписка на журнал «Информатика и образование» на второе полугодие. (N 5, 6). Индекс журнала в каталоге ЦРПА «Роспечать» для индивидуальных подписчиков 70423, для учреждений и организаций 73176 (по безналичному расчету).

ВНИМАНИЮ ВСЕХ ВЛАДЕЛЬЦЕВ БК-0010 (01)

ФИРМА «СНЕК»

- многолетний опыт в области торговли программным обеспечением;
- сотрудничество с лучшими программистами;
 - тысячи программ;
 - гарантия качества;
- самые низкие цены на программы при самых высоких авторских гонорах;
- льготы постоянным клиентам;
- оплата заказа после его получения;
- бесплатный информационный бюллетень для всех клиентов;
- стабильная работа при всех экономических и политических потрясениях.

Мы начинаем выпуск рекламно-информационного бюллетеня «БК-ОБЗОР»

В нем будет печататься подробная информация о новых программах, полезные советы, интервью с лучшими программистами, хит-парады, уроки программирования и т.п.

Мы также проводим конкурсы на лучшую игровую программу.

Подробности - в «БК-ОБЗОРЕ».

Стоимость одного номера 100 руб. (с учетом почтовых расходов).

Тем, кто заказывает у нас программы, бюллетень рассылается бесплатно.

Периодичность примерно 1 раз в 3 месяца. Объем - 17-25 страниц формата А-4.

Господа!

Благодаря вашим заказам мы стали крупнейшим поставщиком программного обеспечения для БК-0010 (01).
Спасибо вам!

Наш адрес: 142407 Московская обл., Ногинск-7, а/я 53.

Телефон для справок, заказов и предложений в Москве: 177-61-64.

Читайте в 1-м и 2-м номерах:

Интервью с Борисом Бортником, Владимиром Савиным, хит-парад, самоучитель программирования на БЕЙСИКе «Мастер», схемы, полезные советы, объявления, реклама, описание более 500 игровых и системных программ.

Адрес редакции: 142403 Московская обл., Ногинск-3, а/я 33.

Для получения 1-го, 2-го и 3-го номеров «БК-ОБЗОРА» необходимо перечислить по 100 рублей за каждый номер на адрес 142407 Московская обл., Ногинск-7, а/я 53.

Квитанцию почтового перевода приложить к письму и указать, какие номера вам выслать.

Те, кто закажет наши комплекты программы, которые мы предлагаем вам ниже, могут получить наш «БК-ОБЗОР» бесплатно.



В. ДЕМЕНТЬЕВ

Сетевая экономическая учебная среда «Капитал»

Идея создания сетевой учебной диалоговой среды (СУДС) для КУВТ УКНЦ и БК-0010 родилась с момента поступления техники в нашу школу. Суть СУДС заключается в том, что учащиеся на своих РМУ, объединенных в локальную сеть, вступают во взаимоотношение друг с другом и с программной средой. Действия учащихся оперативно передаются на другие рабочие места, что позволяет создать единую для всех информационную среду. Использование КУВТ в подобном диалоговом режиме открывает новые нереализованные возможности отечественной техники. Примером использования СУДС может служить сетевая экономическая среда «Капитал».

Сетевая экономическая учебная среда «Капитал» — это программное средство, поз-

воляющее объединить в единую игровую среду сеть из 12 компьютеров УКНЦ. Каждое рабочее место представляет собой пульт, позволяющий учащемуся осуществлять экономические действия от лица «рабочего» или «предпринимателя», вступать с другими участниками игры в отношения «производства и распределения товаров». Цель создания данного ППС — сделать процесс изучения экономики более увлекательным и наглядным.

Практические занятия, заключающиеся в проведении учебных игр, позволяют ученикам глубже усвоить абстрактные экономические понятия, испытать непосредственно на себе действие экономических законов рыночной экономики. Процесс анализа игровых ситуаций, принятия решений и их осуществления

121

РУС
→
←
РУС

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>1</td><td>♠</td><td>ОРЕЛ</td></tr> <tr><td>2</td><td>♣</td><td>ФЕРЗЬ</td></tr> <tr><td>3</td><td>♠</td><td>КОРОЛЬ</td></tr> <tr><td>4</td><td>♣</td><td>ЛАДЬ</td></tr> <tr><td>5</td><td>♠</td><td>КОНЬ</td></tr> <tr><td>6</td><td>♣</td><td>СЛОН</td></tr> <tr><td>7</td><td>♠</td><td>МАСОН</td></tr> <tr><td>8</td><td>♣</td><td>СТРЕЛА</td></tr> <tr><td>9</td><td>♠</td><td>ЩИТ</td></tr> <tr><td>10</td><td>♣</td><td>ДРАКОН</td></tr> <tr><td>11</td><td>♠</td><td>ПЕГАС</td></tr> <tr><td>12</td><td>♣</td><td>МОЛНИЯ</td></tr> </table>	1	♠	ОРЕЛ	2	♣	ФЕРЗЬ	3	♠	КОРОЛЬ	4	♣	ЛАДЬ	5	♠	КОНЬ	6	♣	СЛОН	7	♠	МАСОН	8	♣	СТРЕЛА	9	♠	ЩИТ	10	♣	ДРАКОН	11	♠	ПЕГАС	12	♣	МОЛНИЯ	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top;"> ВЫБОР ДЕЙСТВИЙ Продолжить игру Сменить имя игры СЧИТАТЬ ИГРУ ИЗ ФАЙЛА Записать состояние игры ВЫЙТИ ИЗ ПРОГРАММЫ ПОМОЩЬ </td> <td style="width: 50%; vertical-align: top;"> Сообщения Файл прочитан Игра продолжается Ход 1 место 8 ЖДЯ МАРКЕРА Игра остановлена Дубль сохранен Файл записан </td> </tr> <tr> <td style="vertical-align: top;"> Название игры ВРЕМЯ ИГРЫ Последняя запись Управление переключено </td> <td style="vertical-align: top;"> К6 01:02:23 01:02:23 </td> </tr> <tr> <td colspan="2" style="text-align: center;"> ВЫ УВЕРЕНЫ, ЧТО ХОТИТЕ СЧИТАТЬ ИГРУ ИЗ ФАЙЛА <input type="button" value="ДА"/> <input type="button" value="НЕТ"/> </td> </tr> </table>	ВЫБОР ДЕЙСТВИЙ Продолжить игру Сменить имя игры СЧИТАТЬ ИГРУ ИЗ ФАЙЛА Записать состояние игры ВЫЙТИ ИЗ ПРОГРАММЫ ПОМОЩЬ	Сообщения Файл прочитан Игра продолжается Ход 1 место 8 ЖДЯ МАРКЕРА Игра остановлена Дубль сохранен Файл записан	Название игры ВРЕМЯ ИГРЫ Последняя запись Управление переключено	К6 01:02:23 01:02:23	ВЫ УВЕРЕНЫ, ЧТО ХОТИТЕ СЧИТАТЬ ИГРУ ИЗ ФАЙЛА <input type="button" value="ДА"/> <input type="button" value="НЕТ"/>		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">Капитал</td> </tr> <tr> <td colspan="2" style="text-align: center;">Управляющая программа Новосибирск, 1992 (С) Ерксин А.А., 355119</td> </tr> <tr> <td colspan="2" style="text-align: center;">Для помощи нажмите К1 Поле Рейтинга</td> </tr> <tr> <td style="text-align: center;">♣</td> <td style="text-align: center;">ЛАДЬ 11240</td> <td style="text-align: center;">♠</td> <td style="text-align: center;">2 шт</td> </tr> <tr> <td style="text-align: center;">♠</td> <td style="text-align: center;">КОРОЛЬ 5150</td> <td style="text-align: center;">♣</td> <td style="text-align: center;">2 шт</td> </tr> <tr> <td style="text-align: center;">♣</td> <td style="text-align: center;">ОРЕЛ 5320</td> <td style="text-align: center;">♠</td> <td style="text-align: center;">1 шт</td> </tr> <tr> <td style="text-align: center;">♠</td> <td style="text-align: center;">ЩИТ 7470</td> <td style="text-align: center;">♣</td> <td style="text-align: center;">1 шт</td> </tr> <tr> <td style="text-align: center;">♣</td> <td style="text-align: center;">Молния 7870</td> <td style="text-align: center;">♠</td> <td style="text-align: center;">1 шт</td> </tr> <tr> <td style="text-align: center;">♠</td> <td style="text-align: center;">ФЕРЗЬ 9190</td> <td style="text-align: center;">♣</td> <td style="text-align: center;">1 шт</td> </tr> <tr> <td style="text-align: center;">♣</td> <td style="text-align: center;">Дракон 9310</td> <td style="text-align: center;">♠</td> <td style="text-align: center;">1 шт</td> </tr> <tr> <td style="text-align: center;">♠</td> <td style="text-align: center;">Пегас 10200</td> <td style="text-align: center;">♣</td> <td style="text-align: center;">0 шт</td> </tr> <tr> <td style="text-align: center;">♣</td> <td style="text-align: center;">Стрела 10780</td> <td style="text-align: center;">♠</td> <td style="text-align: center;">0 шт</td> </tr> <tr> <td style="text-align: center;">♠</td> <td style="text-align: center;">Конь 11100</td> <td style="text-align: center;">♣</td> <td style="text-align: center;">0 шт</td> </tr> <tr> <td style="text-align: center;">♣</td> <td style="text-align: center;">Слон 11100</td> <td style="text-align: center;">♠</td> <td style="text-align: center;">0 шт</td> </tr> <tr> <td style="text-align: center;">♠</td> <td style="text-align: center;">Масон 11100</td> <td style="text-align: center;">♣</td> <td style="text-align: center;">1 шт</td> </tr> </table>	Капитал		Управляющая программа Новосибирск, 1992 (С) Ерксин А.А., 355119		Для помощи нажмите К1 Поле Рейтинга		♣	ЛАДЬ 11240	♠	2 шт	♠	КОРОЛЬ 5150	♣	2 шт	♣	ОРЕЛ 5320	♠	1 шт	♠	ЩИТ 7470	♣	1 шт	♣	Молния 7870	♠	1 шт	♠	ФЕРЗЬ 9190	♣	1 шт	♣	Дракон 9310	♠	1 шт	♠	Пегас 10200	♣	0 шт	♣	Стрела 10780	♠	0 шт	♠	Конь 11100	♣	0 шт	♣	Слон 11100	♠	0 шт	♠	Масон 11100	♣	1 шт
1	♠	ОРЕЛ																																																																																																
2	♣	ФЕРЗЬ																																																																																																
3	♠	КОРОЛЬ																																																																																																
4	♣	ЛАДЬ																																																																																																
5	♠	КОНЬ																																																																																																
6	♣	СЛОН																																																																																																
7	♠	МАСОН																																																																																																
8	♣	СТРЕЛА																																																																																																
9	♠	ЩИТ																																																																																																
10	♣	ДРАКОН																																																																																																
11	♠	ПЕГАС																																																																																																
12	♣	МОЛНИЯ																																																																																																
ВЫБОР ДЕЙСТВИЙ Продолжить игру Сменить имя игры СЧИТАТЬ ИГРУ ИЗ ФАЙЛА Записать состояние игры ВЫЙТИ ИЗ ПРОГРАММЫ ПОМОЩЬ	Сообщения Файл прочитан Игра продолжается Ход 1 место 8 ЖДЯ МАРКЕРА Игра остановлена Дубль сохранен Файл записан																																																																																																	
Название игры ВРЕМЯ ИГРЫ Последняя запись Управление переключено	К6 01:02:23 01:02:23																																																																																																	
ВЫ УВЕРЕНЫ, ЧТО ХОТИТЕ СЧИТАТЬ ИГРУ ИЗ ФАЙЛА <input type="button" value="ДА"/> <input type="button" value="НЕТ"/>																																																																																																		
Капитал																																																																																																		
Управляющая программа Новосибирск, 1992 (С) Ерксин А.А., 355119																																																																																																		
Для помощи нажмите К1 Поле Рейтинга																																																																																																		
♣	ЛАДЬ 11240	♠	2 шт																																																																																															
♠	КОРОЛЬ 5150	♣	2 шт																																																																																															
♣	ОРЕЛ 5320	♠	1 шт																																																																																															
♠	ЩИТ 7470	♣	1 шт																																																																																															
♣	Молния 7870	♠	1 шт																																																																																															
♠	ФЕРЗЬ 9190	♣	1 шт																																																																																															
♣	Дракон 9310	♠	1 шт																																																																																															
♠	Пегас 10200	♣	0 шт																																																																																															
♣	Стрела 10780	♠	0 шт																																																																																															
♠	Конь 11100	♣	0 шт																																																																																															
♣	Слон 11100	♠	0 шт																																																																																															
♠	Масон 11100	♣	1 шт																																																																																															

является своеобразным экономическим заданием.

Традиционно учебные экономические игры проводятся по «бумажным» технологиям, когда участники игры собираются в аудитории, ведут между собой переговоры, обмениваются карточками товаров и денег. Подобные технологии проведения учебных игр содержат ряд узких мест. Во-первых, участники не могут узнать своего промежуточного результата. Во-вторых, конечный результат может содержать ошибки, появляющиеся в процессе сбора и обработки информации о состоянии игры. В-третьих, для проведения игр требуется привлекать множество технических работников. В-четвертых, процессом игры трудно управлять. Далее, участники, принимая решения, могут ориентироваться только по своему окружению и не обладают совокупными показателями о состоянии рынка в целом. Процесс игры протекает сравнительно медленно и, как правило, не укладывается в отведенное для занятий время. Каждая игра становится индивидуальной и невоспроизводимой, что не позволяет вернуть ее в промежуточное состояние и продолжить вновь.

Использование сети компьютеров и соответствующих программных средств открывает новые возможности в проведении экономических учебных игр, многие из которых реализованы в описываемом ППС «Капитал». Наиболее существенными моментами в применении сети компьютеров являются: возможность содержать в памяти ЭВМ всю информацию о состоянии игры, существенно увеличить скорость процесса игры, упростить участникам доступ к информации.

Учебная сетевая среда «Капитал» позволяет каждому участнику в процессе игры отслеживать свое место в таблице рейтинга, что является активной обратной связью в оценке эффективности каждого действия. Изменение мест участников в таблице вносит существенный соревновательный момент в игру, повышает активность играющих. В ППС «Капитал» реализована возможность останавливать игру, записывать ее состояние, продолжать вновь или возвращать в промежуточное состояние. Процессом игры полностью управляет один преподаватель. На экране участника после каждого его действия происходит изменение состояния рынка, таким образом решения, принимаемые им, не отстают от реального времени. Оперативность передачи информации по сети позволяет каждому из 12 участников игры за одну минуту совершать до пяти микроэкономических действий и тем самым создавать стихию рынка. Выбор и ввод действий осуществляется с помощью меню.

Идея ППС «Капитал» заключается в том, что участники по ходу игры совершают множество микроэкономических действий типа «купля товаров», «продажа рабочей силы», «ее воспроизведение», «перемещение денег» и др. Неупорядоченные и случайные, они в своей массе складываются в один макроэкономический процесс, в котором проявляется действие экономических законов. Каждое действие игрока здесь подчинено собственным интересам. Чем лучше каждый из играющих понимает их и действует в соответствии с ними (собственные интересы), тем сильнее проявляется в игре влияние экономических законов. Другими словами законы экономики не привносятся в игру со стороны, а, как в жизни, являются прямым следствием поведения ее участников.

По содержанию сетевая экономическая учебная среда «Капитал» соответствует курсу политической экономии общеобразовательной средней школы и рекомендуется для использования в качестве средства обучения в старших классах. Приведем ниже краткое описание правил игры.

Игра «Капитал» моделирует наиболее существенные отношения, складывающиеся между людьми в процессе производства и распределения товаров. К ним относятся: продажа рабочими своей силы, отчуждение от них продукта труда, присвоение предпринимателями прибавочного продукта, конкуренция между товаропроизводителями и между рабочими на рынке труда и пр. В процессе игры учащиеся могут попеременно осуществлять действия, как от лица «рабочего», так и «предпринимателя» в зависимости от того, какая деятельность может принести им больший доход. В начале игры все выступают в роли рабочих и только по мере накопления капитала могут начинать реализовывать предпринимательские функции. К концу игры большинство из них, не выдержав конкурентной борьбы, не в состоянии владеть предприятиями и снова возвращаются в число «рабочих». К трем основным действиям, отражающими ролевую функцию рабочего, относятся «продажа рабочей силы», «покупка товара» и «переход».

Жизненный цикл рабочего (или «сутки») состоит из четырех этапов:

- 1) Поиск места работы.
- 2) Продажа своей рабочей силы (процесс труда).
- 3) Покупка нужных ему товаров.
- 4) Восстановление своей рабочей силы (потребление товаров).

На первом этапе рабочему делаются предложения о продаже его рабочей силы. Если он отказывается от первого, то со-

вершает действие «переход» и получает следующее. В течение суток ему поочередно делается несколько предложений. Причем от каких фирм из ряда, отображенного на экране, ему заранее неизвестно. Отказываясь от очередного, он надеется получить следующее более выгодное, но рискует не получить их больше совсем. В этом случае его рабочая сила пропадает.

Согласившись на предложение, на втором этапе рабочих осуществляет «продажу своей силы». Это действие связано с актом труда, осуществляемым для участника игры мгновенно. В результате его появляется продукт, вид которого определяется типом фирмы. Это могут быть: еда, жилищные услуги, знания или товары общего назначения. За каждый час труда рабочего в игре создается две единицы продукции, например 2 порции еды. Рабочую силу можно продавать по частям на разных фирмах.

На третьем этапе игроки приобретают необходимые им товары. Закупая товары, участники игры учатся распоряжаться ресурсами. В случае если какого-либо товара у «рабочего» не будет, он лишается права восстановить свою рабочую силу и тем самым своего источника дохода до тех пор, пока не приобретет все товары в нужных количествах. Поэтому, запасаясь товарами, рабочий, с одной стороны, обеспечивает себе всегда «крышу над головой» и спасает свои деньги от обесценивания, а с другой, связывая деньги, лишает себя возможности заниматься предпринимательством. Действие «купить товар» осуществляется выбором в меню фирмы и параметром «сколько». Деньги рабочего перечисляются на счет фирмы в соответствии с ценой и количеством.

Для осуществления четвертого этапа необходимо, так же как и для первого, произвести действие «переход». На этом этапе происходит обмен товаров на рабочую силу. Этот процесс как бы связан с потреблением приобретенных продуктов. Продукты исчезают, а рабочая сила восстанавливается. Одна единица продукции обменивается на один час рабочей силы. Какие продукты и в каком количестве обмениваются, определяется структурой потребления, зависящей от «сезона». Осуществляя описанные выше функции, игроки имитируют жизнедеятельность рабочих, формируют спрос на товары и предложение на рынке труда. Соотношения:

1 час рабочей силы → 2 единицы продукции
1 единица продукции → 1 час рабочей силы
для игры являются фундаментальными и обеспечивают расширенное воспроизводство.

Большую роль в «Капитале» играют «банки». В отличие от фирм, производящих товары, «банки» производят деньги. Например,

1 час рабочей силы → 20 долл.

«Банк» можно понимать как золотой прииск, т. е. предприятие, производящее продукт, который нельзя потратить, но можно использовать в качестве всеобщего эквивалента труда и обменивать на него другую продукцию. Если рабочие продают свою силу на фирмах, то на рынке растет масса товаров, если на «банках» — то растет масса денег. Выбытие товаров с рынка происходит за счет потребления их рабочими, а выбытие денег осуществляется за счет предпринимательской деятельности участников игры, в приобретении фирм и вложении средств в основные фонды. Благодаря индивидуальным действиям участников игры массы товаров и денег стихийно регулируются.

Накопив некоторый капитал, участники включаются в предпринимательскую деятельность. Действия «купить фирму», «установить цену товара», «установить цену рабочей силы», «взять товар с фирмы», «положить товар», «снять деньги с фирмы», «положить деньги» относятся к ролевым функциям предпринимателя. Приобретя фирму, участники получают право получать доход от ее деятельности. Например, заплатив за 1 час рабочей силы 17 долл., фирма становится собственником 2 порций еды. При цене еды на рынке в 11 долл. прибавочная стоимость составит $2 \times 11 - 17 = 5$ долл. Но это произойдет только тогда, когда найдется тот, кто продаст фирме рабочую силу, и тот, кто приобретет у нее произведенную продукцию. Таким образом предприниматели формируют спрос на рабочую силу и предложения на товарном рынке, соревнуясь между собой.

Учащиеся в процессе обучения проходят три стадии развития. На первой они учатся выживать как рабочие и не попадать в ситуации, когда у них нет средств, чтобы продлить свое существование. На второй, накопив первичный капитал, включаются в предпринимательскую деятельность, учатся, используя различные ситуации на рынке, извлекать из них максимум прибыли. На третьей они стараются сознательно создавать благоприятные для себя ситуации. Вступают в различные кооперации с другими участниками игры.

После проведения игры с учащимися (процесс труда)

(процесс восстановления)

обсуждаются результаты занятий. Проводятся аналогии между реальным экономиче-

ским поведением людей и игровыми ситуациями. По результатам игр можно проводить отбор ребят, обладающих предпринимательскими способностями.

Описываемое программное средство создано в СШ 130 Новосибирска В. С. Дементьевым и А. А. Брыксиным. В него входят: программа — «пульт» участника игры, программа — «диспетчер», управляющая ее ходом, редактор состояний и исходные файлы.

По итогам конкурса «Программные и технические средства обучения с применением ЭВМ», который проводился в 1992 г. Новосибирским обществом информатики, НИИ ИВТ АПН РАН и внешнеэкономической ассоциацией «Касси» под председательством директора НИИ информатики и ВТ И. М. Бобко, первое место было присуждено сетевой экономической среде «Капитал». Программное средство сертифицировано на уровне ВОИВТ.

*Мы продолжаем строить мост через пропасть, разделяющую
IBM PC и БК.*

Сеть из тридцати БК0011М и центральной IBM PC — это уже реальность!

- Сетевая система SPRUT3 перекрывает возможности предыдущей версии SPRUT2, создает на периферийных компьютерах среду ОС БК-11 и позволяет выполнять программы, работающие под управлением дисковой ОС, такие как редактор EdikM, трансляторы MACRO, C, PASCAL, базы данных и т.д. SPRUT3 имеет экранный интерфейс в стиле NORTON COMMANDER с управлением от «мыши», и другие усовершенствования.
- Электронная картотека FastCard - позволяет создавать, редактировать и сортировать карточки, а также готовить табличные отчеты с выводом на экран, принтер и в файл. Обладает удобным и наглядным пользовательским интерфейсом.
- Редактор баз данных типа dBASE - DBEDIT. Позволяет редактировать созданные на IBM PC или БК-0011М базы данных в формате DBF. Базы данных можно переносить с IBM PC на БК-11М и обратно при помощи программы РСВК.
- Драйвер виртуального диска VM.SYS - организует в расширенной памяти БК-0011М виртуальный диск размером до 120 блоков. Значительно облегчает использование ОС БК-11 и прикладных программ на одном дисковом.
- Экранный калькулятор CL.SYS, имеет большой набор функций, вызывается на экран по нажатию «горячей» клавиши из любой программы.

Мы оказываем услуги по приобретению компьютерных классов, вычислительной и оргтехники, разрабатываем ПО по заказу.



Адрес: 101000, г. Москва, Главпочтамт, а/я 2018.

Контактные телефоны: 230-79-78, 522-43-32.



ИНТЕР  СЕРВЕР

Пакет программ для ПЭВМ «Корвет»

Все категории пользователей этой ПЭВМ заинтересуются предложенным фирмой «Микроэлектронные системы» («МикС») НИИЯФ МГУ пакетом программ. Пакет особенно рекомендуется разработчикам программного обеспечения, так как содержит распространенные и хорошо зарекомендовавшие себя трансляторы языков Си и Паскаль для ОС CP/M-80. Ниже перечисляются некоторые наиболее интересные программы пакета.

Редактор диска DE осуществляет физическое (на уровне дорожек и секторов) редактирование информации на диске. Помогает спасать «испорченные» файлы.

Редактор командной строки DED — расширение возможностей процессора командной строки ОС. Запоминает несколько последних введенных с клавиатуры команд.

Графический редактор ABRIS — редактирование графических изображений (512 × 256 точек). Для ввода рисунка предоставляется 12 «инструментов».

Редактор сложных текстов CHIWORNER — редактирование текстов с формулами и иллюстрациями. Имеются 16 шрифтов и национальные алфавиты. Аналог ChiWriter, файлы документов совместимы.

Музыкальный редактор MUSIC WORLD — среда для создания музыкальных фрагментов. Пригодна для «озвучивания» программ благодаря возможности генерации «музыкального» текста на Ассемблере.

Программы FX и TYPEDOC — печать текстов на Epson-совместимых принтерах с использованием всех их возможностей. TYPEDOC позволяет печатать нестандартными шрифтами и в графическом режиме.

Программа распечатки текстов PRINT — аналогична предыдущим, но позволяет работать с разными принтерами. Возможна печать текста в графическом режиме с иллюстрациями формата редактора ABRIS.

Система программирования на языке Си — пакет «Aztec C» с мощной графической библиотекой.

Система программирования на языке Паскаль — пакет «Pascal MT+» с набором дополнительных библиотек.

Система программирования на языке

Макроассемблер с дополнительными библиотеками и возможностью непосредственной работы с внешними устройствами. В пакет входят стандартные отладчики SID и RESOURCE.

Редактор текстов SUPER TEXT — текстовый процессор с удобной системой редактирования, системами меню и подсказки. Позволяет работать с несколькими текстами одновременно.

Сетевые операционные системы «Корнет» и «Сфера» — работа на PMU в ДОС CP/M-80 с дисковыми РМП при наличии стандартной локальной сети.

Система программирования на языке Паскаль «Express Pascal» — интерактивная среда, аналог Turbo Pascal. Язык близок к Turbo 4.0. Ориентирована на работу в локальной сети.

Англо-русский словарь VCC находит за секунду перевод любого из 20 000 слов.

АОС «Радуга» — инструментальная система создания учебных курсов. Рекомендуется для гуманитарных предметов. Не требует специальных знаний в области программирования.

Электронный справочник по языку BASIC — информация с примерами о любом операторе языка.

Программы TOCOPY и DSDD — копирование файлов и дисков на физическом уровне, перенос операционной системы.

Программа PROFFER — аналог Norton Commander для IBM PC.

Система управления базами данных — стандартный пакет dBASE 2 с примерами. Электронная таблица SUPERCALC.

Программа обучения быстрой печати на компьютере TT.

Обучающая программа LEARNW для быстрого запоминания иностранных слов и их произношения.

Набор исходных текстов игровых программ для системы EXPRESS PASCAL — примеры программ опытных программистов.

Программа IBM осуществляет перенос текстовых файлов с диска формата IBM PC на «Корвет» и обратно.

Библиотека STDLIBC — быстрая и компактная библиотека для Aztec C, заменяющая стандартную LIBC.

УПК готовит к профессиям с использованием ЭВМ

В настоящее время немалое количество учебно-производственных комбинатов (УПК) специализируется в профессиональной подготовке в области вычислительной техники. Школьникам дают такие профессии как оператор ЭВМ, лаборант-программист и т.п.

Опыт показывает, что способности и возможности учащихся в освоении этих профессий существенно различаются по объему и качеству усвоения знаний и умений. Поэтому проблема дифференциации и индивидуализации подготовки является одной из самых важных в деятельности УПК.

Решение этой проблемы, на наш взгляд, состоит в выделении двух основных этапов подготовки, соответственно двум годам обучения.

На первом этапе (10 класс) учащиеся приобретают общепрофессиональные знания и умения: осваивают язык программирования, приобретают навыки работы с клавиатурой, с редакторами и базами данных, простейшие умения работы с операционной системой, с сетью и файлами.

Этому этапу в УПК-3 Черемушкинского района г. Москвы соответствуют виды техники — «Корвет», «Ямаха». Здесь выявляются склонности (научные, художественные, социальные, технические) к профессиям и индивидуальные способности учащихся как в программировании, так и в других видах деятельности.

Первая «примерка» учащихся к тому или иному «профессиональному типу» происходит во время работы с начальным практикумом следующего содержания (состоит из восьми работ):

1.	Тексты	«социальный»
2.	Диалог	
3.	Рисунки	«художественный»
4.	Музыка	
5.	Расчеты	«научный»
6.	Графики	
7.	Исполнитель	«технический»
8.	Накладки	

«Профессиональный тип» проявляется в повышенном интересе и качестве выполнения соответствующих ему работ начального практикума. Учащиеся, обычно, быстро справляются с большим

объемом работы, соответствующей их «профессиональному типу», проявляют к ней повышенный интерес (например, некоторые стараются изменить порядок выполнения работ практикума, чтобы раньше выполнить интересующие их работы), оживляются, активизируются, находят оригинальные решения.

Дальнейшее проявление «профессионального типа» происходит и в ходе работы со вторым практикумом: текстовые, графические, музыкальные редакторы и базы данных.

Цель третьего практикума, наряду с совершенствованием программистских навыков, — дальнейшее выяснение «профессионального типа» и способностей учащихся. Он включает следующие работы:

1. Мультипликация. 2. Моделирование. 3. Тренажеры. 4. Игры.

В мультипликации учащиеся «социального типа» создают движение текста, его мигание, бегущую строку, программы диалога с формированием ответа из движущихся слов или букв и др. Школьники с «художественным типом» организуют движение рисовальных объектов или спрайтов, пишут фоновые мелодии, создают программы, меняющие характер движения графических объектов в зависимости от высоты и длительности звука и т.п. Учащиеся «научного типа» создают графические образы с элементами анимации, различного рода часы, организуют движение по математическим кривым или случайным полжениям и др. Этим ребятам нравится программировать повороты и вращения простых объектов. Учащиеся «технического типа» обычно стремятся организовать движение модели технического объекта, сделать это движение управляемым, организовать взаимодействие движущихся объектов.

В моделировании «профессиональный тип» проявляется, главным образом, в характере создаваемых моделей. У «социального типа» доминирует стремление к воплощению человеческих образов или к созданию моделей с психологическими и социальными параметрами. У «научного типа» в основе моделей лежат достаточно сложные математические закономерности, а сами модели — преимущественно из курсов математики и физики. У «художественного типа» наблюдается стремление к сочетанию в моделях графики и мелодий. Эти ребята стараются создать вариант цветомузыки или смоделировать звуки

голоса. У «технического типа» зачастую моделируются технические установки.

В тренажерах учащиеся «социального типа» предпочитают анализ ответов по методу корней ключевых слов, делают программ. и по угадыванию слов или выражений, иногда организуют генерацию текстов или кроссворды на ЭВМ и т.п. Школьники «художественного типа» делают тренажеры по развитию глазомера и музыкального слуха, по обучению графической и нотной грамотности и т.п. Учащиеся «научного типа» предпочитают работу с формулами, числами, единицами измерения, графиками. Они создают тренажеры по решению задач и примеров и т.п. Школьники «технического типа» делают тренажеры по навыкам вождения автомобиля с соблюдением ПДД, по навыкам приборных измерений, по конструированию механизмов из деталей, по азбуке Морзе, по чтению и рисованию схем и т.п.

В работах, связанных с играми, учащиеся «социального типа» делают макеты игр-единоборств или лингвистических игр. Школьники «художественного типа» предпочитают простые игры с графическим и музыкальным оформлением. «Научный тип» тяготеет к логическим и экономическим играм. «Технический тип» предпочитает игры на стрельбу по мишеням, по различным гонкам, с использованием множества вспомогательных предметов.

Первый год обучения завершается курсовой работой. Она предлагается преподавателем с учетом профессионального типа и способностей ученика, или непосредственно учеником. Как правило, это программы справочно-информационного характера, по обучению языкам программирования, чисто контролирующие программы или тренажеры. Изредка используются моделирующие или игровые программы.

На втором этапе (второй год обучения) школьники делятся на группы в зависимости от способностей. Более способные учащиеся обычно изучают второй язык программирования, углубляют знания об операционной системе, осваивают работу на IBM, овладевают различными тонкостями программирования и переходят к выполнению дипломных работ. Эти работы зачастую связаны с созданием различных версий редакторов или баз данных, с разработкой вспомогательных программ по работе с сетью и файлами на диске, с попытками улучшения «Нортон-системы» в различных аспектах, а также с программами, направленными на решение производственных или научных задач в различных сферах и т.п. Нередко школьниками из таких групп создаются оригинальные версии известных игр или новые игры. В целом можно отметить, что учащиеся этих групп склонны к профессии системного программиста.

Учащиеся обычных групп с самого начала второго года ориентируются на прикладное программирование. Основная область их профессиональной подготовки на этом этапе — создание программ учебного назначения. Эти школьники знакомятся с типо-

выми программными структурами учебного назначения, овладевают приемами создания в учебных программах элементов мотивации, ориентировочной основы деятельности и контроля ее результатов, учатся создавать учебные и инструментальные среды.

Характерные принципы подготовки дипломных работ по программам учебного назначения — групповая организация работ и их комплексность. Как правило, создаются не единичные учебные программы, а их комплексы — практикумы (в среднем — четыре программы) или курсы (20 и более программ). Сначала учащиеся совместно обсуждают типовые программные структуры и особенности конкретного курса. Затем, разбившись на несколько групп, они реализуют в типовых структурах общие элементы. И наконец, школьники переходят к индивидуальной работе над программой. После этого разработанные и отлаженные программы объединяются в практикумы, а те, в свою очередь, — в курсы. Соответственно защита дипломных работ носит групповой характер.

В результате описанного подхода в УПК создаются:

- 1) теоретические и методические разработки, как результат проектирования и опыта работы;
- 2) методические рекомендации по использованию программных средств;
- 3) описание конкретных программ и их комплексов, каталоги созданного и собранного программного обеспечения;
- 4) фонды программ учебного и иного назначения, а также банки вспомогательных модулей программ.

В соответствии с этими направлениями оформляются периодические сборники (альманахи):

1. Концепции и модели деятельности УПК в области ИВТ.
2. Методики использования программных комплексов.
3. Технологии разработки программных комплексов.
4. Оценка эффективности программных комплексов.

Описанная система организации работы УПК, специализирующегося на профессиональной подготовке в области вычислительной техники, позволяет создать базу для перехода к еще одной сфере деятельности УПК данного профиля — организации на базе УПК компьютерного обучения школьников близлежащих школ по различным учебным предметам. Оно может осуществляться преимущественно в форме компьютерных практикумов конкретного целевого назначения и состоящих из них компьютерных курсов. Это достаточно экономичный и организационно простой вариант внедрения компьютерных технологий обучения в школу.

Компьютерные практикумы и курсы по другим предметам, проводимые на базе УПК, будут постепенно формировать основу методической подготовки учителей-предметников в области создания и использования компьютерных технологий обучения.

Справочный листок

128

Со времени выхода в свет № 5—6/92 нашего журнала произошли большие изменения в системе оплаты труда учителей ОИиВТ. На многочисленные вопросы читателей отвечают сотрудники отдела труда и заработной платы МО РФ.

Та часть постановления Министерства труда Российской Федерации № 16 от 04.06.1992 г. «О размерах надбавок и доплат работникам образования», в которой говорилось о 5 %-ной доплате к ставке учителя ОИиВт за обслуживание вычислительной техники за каждый работающий компьютер, в настоящее время не действует в связи с введением с 01.12.1992 г. единой тарифной сетки.

Письмом Министерства образования Российской Федерации от 12.01.1993 г. № 10/32-Т предложено ввести вместо ранее существовавших доплат должность (часть должности) для обслуживания электронно-вычислительной техники с оплатой в установленном порядке. Для этого могут быть введены должности инженера, лаборанта или техника по обслуживанию вычислительной техники.

Учитель ОИиВТ в пределах основного рабочего времени может одновременно совмещать свою работу с работой лаборанта этого кабинета. Конкретный размер доплаты учителю за совмещение должности лаборанта определяется руководителем учреждения по согласованию с профсоюзным комитетом в зависимости от объема выполняемой работы, но не может превышать фонда заработной платы, предусмотренного по должности лаборанта (фонд заработной платы определяется с учетом доплаты за вредные условия труда).

Читатель Квитко Т. Н. (г. Владимир) спрашивает:

«Полагается ли учителям ОИиВТ надбавка за вредность работы в компьютерном классе?»

На основании письма Министерства образования Российской Федерации от 12.01.1993 г. № 10/32-Т доплата за работу в неблагоприятных условиях производится в размере до 12 % к ставке заработной платы (окладу).

Никитенко Л. В. (с. Амвросиевка) спрашивает:

«При определении разряда оплаты труда по ЕТС включается ли учителю ОИиВТ время работы инженером-программистом в стаж педагогической работы?»

Да, время работы в должности инженера-программиста засчитывается учителю ОИиВТ в педагогический стаж в соответствии с приказом Госкомитета СССР по народному образованию № 400 от 08.06.1990 г., приложение № 6.

Редакция журнала «Информатика и образование» благодарит читателей за присланные письма и обещает отвечать на все вопросы, привлекая для ответов соответствующих специалистов.

Индивидуальное обучение –
гарантия отличного образования!

Индивидуальное обучение –
мечта преподавателя и учащихся!

К КОЛЛЕДЖ

Программно-методический комплекс

«Информатика»

приблизит Вас к этой мечте!

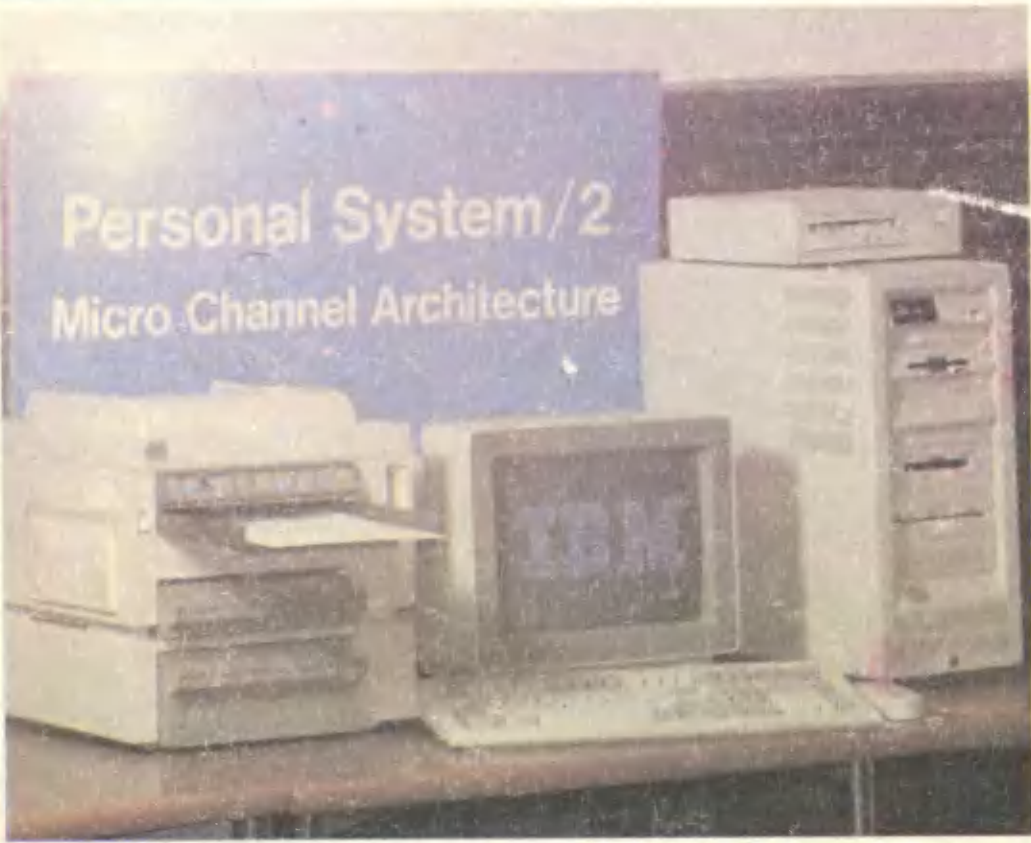
1. Учебная программа - поможет решить проблемы планирования занятий.
2. Автоматизированная обучающая система (АОС) *„ЭКОРМ“* - позволит после 4-8 часов работы управлять классом ПЭВМ.
3. АОС *„КОЛЛЕДЖ“* - поможет Вашим учащимся освоить курс «ОИ и ВТ» и проверить полученные знания, ответив на контрольные вопросы и сдав зачет. А Вам останется согласиться, что «учение с увлечением» - это реальность!
4. Сборник дидактических материалов - предоставит каждому из Ваших учащихся карточки-задания (30 вариантов) для 20 лабораторных работ и методические рекомендации по проведению этих работ для Вас.
5. Библиотека программ на BASIC - иллюстрирует и комментирует работу операторов и структуру языка; рассылается по сети.
6. Пакет прикладных программ - позволит Вашим учащимся развить и реализовать свои творческие возможности при работе на компьютере.
7. Уроки с программируемым микрокалькулятором.

ПМК *„Информатика“* решает Ваши проблемы
и реализует Ваши идеи!

ПМК *„Информатика“* реализован для УК-НЦ,
УК-НЦ01.01, КУВТ86, ДВК, IBM

Наш
адрес:
107005,
Москва
Волховский
пер., 11
Фирма «Колледж»
телефон: 267-7058

Наши перспективы - создание подобных комплексов по другим школьным предметам. Если у Вас есть интересные методические наработки, но не хватает времени и возможностей, фирма «Колледж» готова помочь Вам создать законченный программный продукт.



Personal System/2
Micro Channel Architecture