

ИНФОРМАТИКА И ОБРАЗОВАНИЕ

6 1993



**СПОНСОРЫ ЭТОГО НОМЕРА – ВЕДУЩИЕ ФИРМЫ РОССИИ
В ОБЛАСТИ ИНФОРМАТИЗАЦИИ ОБРАЗОВАНИЯ:**

Научный Центр Программных Средств Обучения

Генеральный директор А. А. Зубченко

Фирма AIST AB

Генеральный директор В. А. Урнов

«... Там, где роль наглядности велика, – исторические реалии, флора и фауна климатических зон, карты, процессы жизнедеятельности организма, представленные в динамике, химические реакции, сопровождающиеся видимыми изменениями, а также произношение звука, интонация во фразе, анализ музыкальных стилей и так далее – вопрос о применении информационных технологий мультимедиа нами принципиально решен, и здесь MultiVision PRO трудно переоценить.»



**Основная часть демонстрационного устройства, входящего в комплекс
MultiVision PRO**

**На первой стороне обложки: перспективная модель уникального демонстрационного
устройства DESKTOP ПРОЕКТОР – составная часть следующей версии комплекса
MultiVision PRO**

О комплексе MultiVision PRO читайте в статье «Мы поможем вам успеть в XXI век»

ИНФОРМАТИКА И ОБРАЗОВАНИЕ

ИНФО
6'93

Научно-методический журнал
Учрежден Министерством
образования РФ
и коллективом редакции

Издается с августа 1986 г.
Выходит шесть раз в год

СОДЕРЖАНИЕ

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

БОЛОТОВ В. А.
БРЕДИХИН Г. А.
ВАСИЛЬЕВ Б. М.
ЗАЙДЕЛЬМАН Я. Н.
ЗУБЧЕНКО А. А.
КИСЕЛЕВ Б. Г.
КОРОЛЕВ В. А.
КРАВЦОВА А. Ю.
КРАСНОВ А. Я.
КУЗЯКИН А. П.
КУРНЕШОВА Л. Е.
ЛАПЧИК М. П.
ПАХОМОВА Н. Ю.
САМОВОЛЬНОВА Л. Е.
САПРЫКИН В. И.
СМЕКАЛИН Д. О.
УВАРОВ А. Ю.
УГРИНОВИЧ Н. Д.
ФУРСЕНКО А. И.
ХОРОШИЛОВ В. О.
ХРИСТОЧЕВСКИЙ С. А.
ЧУРИКОВ П. А.

ОБЩИЕ ВОПРОСЫ

- Ершов А. П. О человеческом и эстетическом факторе
в программировании 3
Программа информатизации образования в Российской
Федерации на 1994—1995 гг. 8

МЕТОДИКА

- Лесневский А. С., Белошапка В. К. Требования к знаниям
и умениям школьников по информатике 25
Степанов М. Е. Единый методический подход к построению
графических изображений 30
Пахомова Н. Ю. Деловые игры на уроке
информатики 43
Познайте мир с нами! (Интервью с научным руководителем
КУДИЦ Б. Г. Киселевым) 47

ИНФОРМАТИКА В МЛАДШИХ КЛАССАХ

- Первин Ю. А. Учительские семинары по
Роботландии 49
Сякина М. В., Первин Ю. А. Исполнитель Тяни-Толкай
на уроках Роботландии 59
Ломко Е. Б. Развивающие игры фирмы «Никита» 68

РЕДАКЦИЯ

Первый зам. гл. ред.

Кравцова А. Ю.

Зам. гл. ред.

Васильев Б.М.

Отдел методики

Луцкая Н.А.

Отдел ВТ

Усенков Д.Ю.

Отдел рекламы

Васильева Н.А.

Производственный отдел

Иванов В.Л.

Компьютерная верстка

Иванова Т.В.,

Панченко О.Н.

Экономический отдел

Бородаева З.В.

Технический редактор

Луговская Т.В.

Корректор

Антонова В.С.

НОВЫЕ ПРОЕКТЫ

Урнов В. А., Крылова Е. С. Мы поможем вам успеть в XXI век

73

ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР

Байбаков В. О. Организация интерфейса программы на языке QUICK BASIC

83

КЛУБ «КОРВЕТ»

Никитин С. А. Секреты «Корвета»

93

КЛУБ УКНЦ

Брусенцев В. А., Козин А. В. IBM на УКНЦ

115

Баранов О. Е. Модем для УКНЦ

120

НАМ ПИШУТ

Горвиц Ю. М. Международный детский компьютерный лагерь-школа

123

Златопольский Д. М. Кроссворд

126

Положение для постоянного определения или оценки успехов в науке

128

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

Телефон: (095) 208-30-78

Факс: (095) 208-67-37

При перепечатке материалов ссылка на журнал обязательна.

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

Подписано в печать с оригинал-макета 16.03.94. Формат 70x100 1/16. Бумага офсетная. Печать офсетная.
Усл.печ.л. 10,40. Усл. кр.-отг. 11,70. Уч.-изд. л. 13,5. Тираж 24 000 экз. Заказ 1947.

Цена по подписке:

для индивидуальных подписчиков 120 руб. (индекс 70423);

для предприятий и организаций 200 руб. (индекс 73176).

В розницу цена договорная.

© «Информатика и образование», 1994

Мы продолжаем публиковать материалы из творческого наследия академика А. П. Ершова. Эта статья была напечатана в разное время в ведущих научных отечественных и зарубежных изданиях.

А. П. Ершов

О ЧЕЛОВЕЧЕСКОМ И ЭСТЕТИЧЕСКОМ ФАКТОРАХ В ПРОГРАММИРОВАНИИ

Кому-нибудь может показаться странным намерение опубликовать в научном журнале статью по столь субъективному вопросу. История науки, однако, показывает, что в определенные периоды эстетические, организационные и вообще внешние по отношению к техническому содержанию научной дисциплины факторы вносили иногда решающий вклад в формирование и развитие данной дисциплины. Сейчас, когда программирование как наука и как профессия вступает в период своего самоопределения, анализ человеческих факторов в программировании представляется автору актуальным.

Дело в том, что, несмотря на то, а может быть, и благодаря тому, что программирование признается сейчас ключевым моментом в расширении и углублении сферы применения ЭВМ, для программистов наступают трудные времена. Объем и сложность программ возрастают непропорционально по отношению к зарплате. Романтический ореол непостижимости этой профессии, если он когда-либо и существовал, начинает меркнуть. На Западе софтверхаусы тают как вчерашний снег, а программисты начинают пополнять армию безработных. Оспаривается даже само стремление рассматривать программистов как профессионалов особого рода. Главное же — это то, что вольная армия программистов постепенно попадает в «плен» к администраторам и руководителям, которые стремятся сделать труд программиста планируемым, измеряемым, однородным и обезличенным.

У читателя не должно создаться впечатления, что автор считает эту тенденцию неправильной. Недостаточная эффективность труда программистов является, может быть, главной причиной существующего разрыва между потребностями и возможностями успешного применения ЭВМ.

С этих позиций следует согласиться, что как профессия программирование еще не достигло своей зрелости. На Западе характерным свидетельством этого в течение последних лет была волна мелкого бизнеса, связанного с так называемыми софтверхаусами, соорудившимися в течение нескольких недель группой толковых программистов как правило, покинувших большую организацию, в которой они получили начальный опыт. В большинстве случаев мотивом для такой инициативы была жажда наживы, полудетское желание избавиться от излишней опеки, конечно, в сочетании с некоторой интересной и полезной идеей в области разработки софтвера. Однако в дальнейшем жизнеспособными оказались лишь такие коллективы, в которых этот партизанский дух быстро заменялся режимом экономии, иерархией отношений, жесткой дисциплиной — словом, всем тем, что в свое время вытолкнуло их из родительского дома. В качестве шутки можно заметить, что вся эта история напоминает сказку о трех поросятах: братья-программисты в конце концов собрались в крепком софтверхаусе, но лишь после того, как первые два были унесены волчьим ветром беспощадной коммерции.

Следует отметить, что хотя и в других проявлениях, но аналогичные явления наблюдались и у нас, когда несколько лет назад бездумный оптимизм и наивная вера во всемогущество

машины в некоторых проектах заменяли собой трезвый расчет, крепкую организацию и качественное составление программ.

Таким образом, подчинение программирования промышленным методам работы — это неизбежный факт. Автор считает, однако, что эта тенденция должна быть сбалансирована встречной инициативой, состоящей в том, что программист должен найти некоторую систему внутренних ценностей в своем деле, обладание которой позволит ему легче ассимилировать индустриальные методы работы, где надо — преодолевать их.

Автор убежден в том, что эта система ценностей в программировании объективно существует, однако осознана не до конца, известна не всем и поэтому требует распространения и защиты. Эта система имеет много компонент, пожалуй, самая важная из них — это профессиональный статус программиста (надо заметить, что о программистах здесь говорится в широком смысле, причисляя к ним и системных аналитиков), но автору в данный момент больше хочется сказать об эстетической или об эмоциональной стороне программирования, причем не только о том, что вознаграждает программиста, когда он выходит со своим продуктом к потребителю, но и о том, что составляет его нравственную опору, когда он остается наедине с программой или машиной.

Программирование становится массовой профессией. Однако надо иметь в виду, что сейчас это, пожалуй, самая трудная из всех массовых профессий, причем, к сожалению, эта трудность не признана в должной мере.

Трудность заключается в том, что именно программисты непосредственно упрутся в пределы человеческого познания в виде алгоритмически неразрешимых проблем и глубоких тайн работы головного мозга.

Трудность состоит в том, что собственный стек программиста должен быть глубиной не в 5—6 позиций, как это обнаружили психологи у среднего человека, а той же, что и стек в его очередной задаче, подлежащей программированию, плюс еще две-три позиции.

Трудность также и в том, что программист должен обладать способностью первоклассного математика к абстракции, логическому мышлению в сочетании с эдисоновским талантом сооружать все что угодно из нуля и единицы. Он должен сочетать аккуратность бухгалтера с пронизательностью разведчика, фантазию автора детективных романов с трезвой практичностью экономиста. А кроме того, программист должен иметь вкус к коллективной работе, понимать интересы пользователя и многое другое.

В работе эта трудность может быть преодолена только путем большого эмоционального напряжения, требующего от программиста особого самосознания и внутренней позитивной установки. Понимание этой установки необходимо для тех, кто управляет программистами, и в особенности для тех, кто их воспитывает и обучает. В качестве примера можно перечислить некоторое количество организационных альтернатив или просто трудных вопросов, правильно разрешать которые можно только с полным учетом обсуждаемых факторов:

Возможна и нужна ли организация разработки софтвера по принципу конвейерной линии?

Кого и почему труднее найти для реализации софтверного проекта — руководителя или исполнителя?

Как сочетать элитарность системного программирования с его массовостью?

Как воспитывать программиста — через мировоззрение (университет) или путем профессиональных навыков (технический институт)?

Что такое индивидуальные способности в программировании, специфичны ли они и нужны ли?

Можно и нужно ли отделять проектирование большой программы от ее изготовления?

Эти вопросы являются частью общей проблемы, поэтому сделаем лишь частные комментарии при попытке связать их постановку с анализом человеческого фактора в программировании.

О конвейере

В каком-то смысле конвейер является дьявольским изобретением. Поднимая продуктивность на небывалый уровень, он в то же время в максимальной степени превращает человека в придаток машины. Конвейерный метод в программировании может либо убить интеллектуальную компоненту в труде программиста, либо вызвать неврозы из-за противоречия между монотонностью и трудностью работы. Представьте себе человека, обязанного 8 часов в день, 5 дней в неделю, 50 недель в году решать одни кроссворды, и вы поймете, что такое программист, специализирующийся, например, на написании редактирующих программ. Одним словом, раскрепление людей по элементарным операциям в многомодульной системе — далеко не простая задача.

О руководителях и исполнителях

Не торопитесь ставить руководителя на первое место, объясняя, что по определению руководителя найти или создать труднее. Давайте подумаем: почему сплошь да рядом руководитель проекта предпочитает начинать с молодыми специалистами, кончившими университет два-три года назад, нежели с людьми, чей стаж работы превышает пять лет? Не потому ли, что мы предпочитаем использовать чистый лист и пластичность молодого человека, нежели преодолевать пассивное сопротивление более зрелого и менее ясного для нас 33-летнего главы семейства? Но это, в частности, означает, что мы не умеем гармонично развивать профессиональные достоинства исполнителя так, чтобы они не падали с возрастом и были полезны не только для руководителя, но и для него самого и его будущих начальников.

Элитарность программистов представляется автору очевидной и в этом виде является серьезным вызовом человечеству в целом, причем можно надеяться, что вызов будет принят и преодолен. Эта мысль будет расшифрована несколько позднее.

Мировоззрение и профессионализм

Проблема, конечно, не только в том, чтобы объективно оценить требуемое соотношение кандидатов наук и дипломированных инженеров-программистов, хотя вокруг возникает изрядное количество всем известных кадровых проблем. Суть проблемы в том, чтобы признать, что программирование требует от человека несколько особого взгляда на мир, его потребности и эволюцию, особой моральной подготовленности к своему долгу. Программист — это солдат научно-технической революции и как таковой должен обладать революционным мышлением.

Теперь мы подходим к тому, чтобы сформулировать центральный тезис статьи. Он состоит в утверждении, что программирование обладает богатой, глубокой и своеобразной эстетикой, которая лежит в основе внутреннего отношения программиста к своей профессии, являясь источником интеллектуальной силы, ярких переживаний и глубокого удовлетворения. Корни этой эстетики лежат в творческой природе программирования, его трудности и общественной значимости.

Здесь, прежде чем продолжить основную мысль, автор хотел бы подчеркнуть важность внутреннего отношения человека к своему делу. Сейчас идет много споров о том, является ли программирование специфической профессией. Это не отвлеченный спор, а дискуссия, результат которой имеет прямые организационные, юридические и образовательные последствия. Главным залогом успешного исхода этой дискуссии должны быть прежде всего самосознание и способность к взаимопониманию тех, кто относит себя к программистам. Известная пословица «Рыбак рыбака видит издалека» должна найти свою интерпретацию в программистской среде.

Выделить эстетическую сущность любого вида профессиональной деятельности очень непросто. Она по своей сути реализуется в субъективных категориях и глубоко сплетается с эстетическим кодексом профессии, с ее техническим содержанием и юридическим статутом. Поэтому перечисление эстетических компонент программирования в этой статье также будет носить субъективный и очень предварительный характер.

Сначала сделаем некоторые замечания, отражающие внутреннюю природу программирования.

Творческая и конструктивная природа программирования не требует особых доказательств. Автор хотел бы высказать, быть может, более спорную мысль, что в своей творческой природе программирование идет намного дальше большинства других профессий, приближаясь к математике и писательскому делу. В большинстве других профессий мы лишь «приручаем» при помощи сил природы те или иные физические или биологические явления, не обязательно постигая их сущность. В программировании же мы в некотором смысле идем до конца. Один из тезисов современной теории познания — «мы знаем что-то, если можем это запрограммировать» — очень выпукло характеризует этот максимализм нашей профессии.

Другим очень важным эстетическим принципом программирования является его высочайшая требовательность к законченности продукта. Конечно, это характерно для многих инженерных профессий. Однако программирование и здесь идет дальше. Хотя в мультимиллионных программных конгломератах это свойство почти исчезает, однако на уровне индивидуальной работы всегда существует поразительный контраст между почти сделанной и полностью сделанной работой. Эта стопроцентность программирования — источник его трудности и в то же время глубочайшего удовлетворения работающей программой.

Машина, снабженная программой, ведет себя разумно. В этот кульминационный момент программист сознает, что его программа, получая самостоятельную жизнь, материально воплощает его интеллектуальные усилия, становящиеся отныне общим достоянием. Это торжество интеллекта, наверное, самая сильная и самая специфическая сторона программирования.

В отношении к машине у добросовестного программиста есть еще одна особенность. В некотором смысле он относится к ней, как хороший жокей к своей лошади. Зная и хорошо понимая возможности машины, он никогда не позволит себе компенсировать леность ума беззаботной тратой ресурсов ЭВМ. Это чисто эстетическое отношение к делу является самым эффективным предохранителем против бездумной «пессимизация» софтвера, которая иногда сводит на нет эффективность использования машины.

Другую часть эстетической сущности программирования составляют такие его компоненты, которые связаны с социальной, или общественной, функцией программирования. Всякий раз, когда мы рассматриваем социальное явление большого масштаба (а появление и использование ЭВМ, безусловно, является таковым), мы должны поискать некоторые широкие исторические аналогии, которые могут дать какую-то опору для экстраполяции и предвидения. О том, что ЭВМ принесли с собой научно-техническую революцию и связанную с ней индустриализацию умственного труда, уже говорилось. В этом месте хотелось бы провести еще одну аналогию, которая имеет более прямое отношение к профессии программиста. Разработка и распространение софтвера во многом напоминают то, что произошло в результате появления книгопечатания. Как книги накапливают внешний образ мира в глазах их авторов и позволяют воспроизвести процесс его познания, так и программы и банки данных накапливают информационную и операционную модели мира и позволяют не только воспроизводить, но и предсказывать его эволюцию, давая тем самым небывалую власть над природой.

Быть сейчас хорошим программистом — это такая же привилегия, как быть грамотным человеком в XVI веке. Эта привилегия дает право программисту ожидать аналогичного признания и уважения со стороны общества. К сожалению, эти ожидания не всегда оправдываются. Следует, однако, заметить, что осуществление такого признания требует работы с обеих сторон. В частности, для программиста необходимо следование одному этическому принципу, который носит общий характер для всякого профессионала, но имеет специальную интерпретацию для программиста. Несколько упрощенно имеют место три варианта: работа ради работы, работа ради денег, работа ради цели.

В системе координат программиста первые два мотива стоят на первом плане, хотя в абсолютной системе координат имеет значение лишь третье. В связи с этим надо всегда помнить, что программист сможет достичь мощной гармонии с обществом только в том случае, если лояльность той цели, в достижении которой его программа является лишь частью, станет его внутренней установкой.

Говоря об общественной функции программирования, нельзя не заметить, что на пути к реализации этой функции лежит одна нерешенная техническая проблема — обеспечение аккумулятивного эффекта программирования. Это очень сложная, но абсолютно необходимая для решения проблема. Спектр мнений о ней бесконечен. Одни говорят, что сейчас работают только считанные проценты составленных программ, другие считают что ОС/360 — это уже практически бессмертный комплекс программ. Возвращаясь к теме статьи, хочется сказать, что предоставление программисту перспективы длительного и стабильного использования продукта его труда окажет решающее воздействие на его профессиональное самосознание.

Автор хотел бы теперь с позиций только что сделанных утверждений завершить обсуждение ранее перечисленных альтернатив и трудных проблем.

Об индивидуальных способностях в программировании

Нам необходим образ идеального программиста. Конечно, это будет мифическая личность. Но кто сказал, что нам не нужны мифы и сказки о программистах? Каждый из нас должен хоть раз в жизни видеть или хотя бы слышать о чудо-программисте, из программы которого нельзя убрать ни одной команды или который пишет тысячу команд в день либо обнаруживает ошибку при исходном шансе один к миллиону и т. д. Человеку свойственно искать ориентиры и примеры. Именно с этих позиций, по-видимому, следует решать спор о пресловутых «примадоннах» в командах программистов. Объявлять их нежелательными — это по крайней мере близорукость или зависть к их исключительным качествам. Автору посчастливилось в жизни

встретить несколько таких примадонн от программирования, которые при всей их индивидуальности и даже экстравагантности вносили неоценимый вклад в работу группы, в особенности в трудных ситуациях. Так что надо признавать и полностью учитывать весьма широкий диапазон индивидуальных способностей к программированию.

О разделении проектирования и изготовления софтвера

Налицо двойственное отношение к этому вопросу. С одной стороны руководители, ответственные за долговременные проекты, и многие другие ищут пути к формализации этапов разработки и передаче проекта из одних рук в другие. С другой, само дело отчаянно сопротивляется такому разделению. По-видимому, правильное решение этого вопроса невозможно без учета человеческого фактора и эстетической потребности, препятствующей тому, чтобы заниматься реализацией чужих идей или не видеть самому овеществления своей идеи. Отдавать технический проект в другие руки — то же самое, что посылать своих детей в интернат.

В заключение вернемся к тезису об элитарности программирования и его будущем. Наша апологетика на первый взгляд подчеркивала исключительный, особый характер программирования и его предельные требования к человеческим возможностям. Эта требовательность и образует тот самый вызов человеку, о котором говорилось выше. Во время пребывания в 1970 г. в Соединенных Штатах на автора произвели очень большое впечатление новые идеи профессоров Массачусетского технологического института Марвина Минского и Сеймура Пейперта об обучении детей. Они выбросили в корзину расхожее представление о том, что дети учатся бессознательно, методом подражания. Они доказывают, что человек чему-то научается только в том случае, если у него в голове складывается блок-схема действия, выделены подпрограммы и проложены информационные связи. Профессор Пейперт всегда обратил автора в свою веру на примере жонглирования двумя мячами, когда, апеллируя к его способностям программиста, он за десять минут научил его тому, чего бы он сам не сделал и за несколько часов.

Таким образом, человек неизмеримо усилит свой интеллект, если сделает частью своей природы способность планировать свои действия, вырабатывать общие правила и способ их применения к конкретной ситуации, организовывать эти правила в осознанную и выразимую структуру, — одним словом, сделается программистом.

Когда-то возможность читать и писать считалась делом избранных. Сейчас, в эпоху грамотности, на что потребовалось 1000 лет, мы выделяем новую избранную категорию людей, которые становятся посредниками между человечеством и информационной моделью мира, упрятанной в машины. Сделав искусство программирования общим достоянием, мы лишимся своей элитарной исключительности перед лицом повзрослевшего человечества. Это ли не высший эстетический идеал для нашей профессии?

Для того чтобы осуществить такой скачок, человечеству понадобится много меньше чем 1000 лет, однако сейчас мы еще очень далеки от этого. Нас окружают более прозаические проблемы, требующие немедленных действий. Однако внутренний мир каждого человека, в том числе и скромного эм-эн-эса или инженера, читающего толстое руководство по программированию или ищущего нужную клавишу за терминалом, хранит в себе неисчерпаемую глубину мыслей, желаний и переживаний. Автор глубоко убежден, что дело, которым занимается программист, требует и от его коллег, и тем более от его руководителей существенно большего понимания мотивов к выполнению его профессионального долга и перспектив его жизненного пути.

Мы назвали ряд актуальных проблем, связанных с человеческим фактором в программировании. Пожалуй, самая главная не была названа. Поколения людей меняются значительно медленнее, чем поколения машин. Автор хотел бы спросить у своих коллег-руководителей, знают ли они, как делать, чтобы программист в возрасте свыше 50 лет был бы не меньше полезен, нежели 30-летний. Через 30 лет таких программистов в мире будет миллион. Пожалуй, честно будет сказать, что сейчас у нас еще нет надлежащего подхода к тому, как ассимилировать ветеранов в современных условиях изменчивости и нестабильности, сделав тем самым профессию программиста пожизненной и дающей человеку комфортабельное ощущение общественной и профессиональной полезности.

ПРОГРАММА ИНФОРМАТИЗАЦИИ ОБРАЗОВАНИЯ В РОССИЙСКОЙ ФЕДЕРАЦИИ на 1994—1995 гг.

1. Общие положения и основные направления информатизации образования

Данный документ разработан в соответствии с Федеральной Программой развития образования в Российской Федерации, в которой информатизация образования выделена в качестве целевой программы (проекта) и подчеркнута приоритетность этого направления для развития всей системы образования России.

Информатизация означает для образования значительно больше, чем просто внедрение в учебный процесс нового содержания и новых технологий. Уровень развития информатизации характеризует в современном мире уровень развития государства, без мощной информационной инфраструктуры невозможно создать цивилизованный рынок товаров и услуг, обеспечить решение задач социальной сферы, войти в мировое сообщество в качестве полноценного партнера.

Успешность осуществляемых в Российской Федерации социальных и экономических реформ связана с достижением прогресса в области информатизации. Но без прочного фундамента, заложенного в сфере образования, никакие, даже самые выдающиеся отдельные научные открытия не сделают наше общество действительно информационным.

Для того чтобы система образования смогла готовить граждан информационного общества, она сама должна стать информационной. Невозможно учить тому, чего не знаешь и не умеешь сам. При этом информатизация не должна делаться для образования обязательной повинностью, а войти в него органично, способствовать его совершенствованию, открыть дополнительные возможности развития.

Потенциальные возможности информатизации в совершенствовании образования чрезвычайно высоки. Современные средства информатизации позволяют: накопить и сделать легко доступными преподавателям и учащимся громадные объемы учебно-методических материалов; обеспечить высокую наглядность учебного материала; осуществлять коллективную работу учащимся, находящимся в разных городах, а учащимся, сидящим за одним столом, работать по индивидуальным программам. Тем самым информатизация открывает широкий простор для творчества как преподавателей, так и учащихся, что является одной из важнейших задач образовательной реформы в Российской Федерации.

Для того чтобы реализовать эти потенциальные возможности информатизации, необходимо:

- обеспечить образовательные учреждения средствами вычислительной и организационной техники, которые бы позволили применение новых информационных технологий (НИТ) в обучении;
- подготовить всех работников сферы образования к использованию НИТ;
- разработать вариативные учебно-методические материалы (в форме компьютерных программ, печатной, аудио- и видеопродукции), позволяющие обеспечить изучение НИТ, а также применение НИТ в качестве средства обучения во всех дисциплинах для образовательных учреждений всех типов и уровней;
- создать инфраструктуру, обеспечивающую доступность образовательной информации всем образовательным учреждениям;
- создать в сфере образования условия, стимулирующие создание и применение инноваций;
- создать действенную систему контроля качества работы образовательных учреждений, их способности удовлетворить образовательные потребности населения.

Несмотря на сложность указанных задач информатизации образования, пути их решения к настоящему времени стали достаточно ясны.

Информационная инфраструктура в сфере образования может быть реализована на основе уже существующих в Российской Федерации телекоммуникационных сетей при условии создания и

подключения к ним банков образовательной информации. Ведение указанных банков, помощь образовательным учреждениям в работе с сетью и в использовании телекоммуникаций в качестве средства обучения требуют создания специальных центров (узлов сети), но эта деятельность вполне может осуществляться уже существующими учреждениями подготовки и переподготовки педагогических кадров.

Обеспечение образовательных учреждений вариативными учебно-методическими материалами, ориентированными на применение НИТ, может быть реализовано следующим образом.

Прежде всего, на основе уже имеющихся работ и минимально необходимых новых разработок образовательным учреждениям предоставляется возможность реализации рекомендаций Базисного учебного плана.

В дальнейшем создание новых продуктов должно стимулироваться не столько за счет финансирования из государственных программ, сколько благодаря переходу к рыночным отношениям. Для этого требуется обеспечить охрану авторских прав, а также создать систему информирования образовательных учреждений о новых разработках. Последнее может быть реализовано путем создания соответствующей «электронной биржи» — баз данных, подключенных к образовательной телекоммуникационной сети.

Первым шагом на пути совершенствования управления образованием должно стать создание и введение в действие государственных стандартов в области информатизации образования. Без введения соответствующих нормативов контроль качества работы образовательных учреждений просто невозможен.

Необходимо также введение государственной сертификации аппаратных и программных средств, рекомендуемых к использованию в учебном процессе. Экспертиза указанных продуктов может осуществляться соответствующими экспертными советами на местах.

В дальнейшем совершенствование управления образованием может быть достигнуто за счет большей информированности органов управления о состоянии дел в подведомственных образовательных учреждениях, а самих образовательных учреждений — обо всех нормативных документах, другой организационно-управленческой информации. Использование телекоммуникационной сети должно позволить не только автоматизировать процесс пересылки, получения и обработки информации, но и использовать опыт коллег для повышения качества работы.

Обеспечение системы образования кадрами, готовыми к применению НИТ в своей работе, может осуществляться за счет подготовки молодых специалистов в педвузах (требуется введение новых специальностей и осуществление подготовки в области НИТ всех студентов) и создания на базе уже имеющихся учреждений повышения квалификации центров информатизации, где все работники сферы образования могли бы получить необходимую им подготовку. В условиях единого информационного пространства эта задача упрощается, так как перед учреждениями подготовки и переподготовки кадров открывается возможность большей кооперации с аналогичными учреждениями, а следовательно и возможность специализации.

Наибольших финансовых затрат в области информатизации образования требует оснащение образовательных учреждений средствами информатизации. Учитывая трудное материальное положение отрасли, трудно надеяться, что этот процесс может идти быстрыми темпами. Поэтому чрезвычайно важно избегать лишних затрат, приобретая морально устаревшую технику или же технику разных типов. Программно и аппаратно несовместимые компьютеры требуют существенного увеличения затрат на содержание большего числа ремонтных центров, не позволяя объединения в единую образовательную сеть, требуют создания для каждого типа собственного учебного программного обеспечения, чем увеличивают стоимость программных продуктов и препятствуют насыщению ими образовательного рынка.

Работы в указанных направлениях ведутся в Российской Федерации уже несколько лет. Научно-исследовательские институты, авторские коллективы создали определенный научный и учебно-методический задел. Ряд организаций создает свои образовательные базы данных, фрагментарно действуют образовательные телекоммуникационные сети. В отдельных регионах реализуются региональные Программы информатизации образования, создаются учебно-методические центры информатизации, центры по ремонту и обслуживанию вычислительной техники, фонды педагогических программных средств.

Вместе с тем информатизацию образования, создание единого информационного образовательного пространства Российской Федерации невозможно осуществить усилиями отдельной организации или региона. Даже общими усилиями ее не удастся осуществить за 2 года или 5

лет, на которые обычно рассчитаны программы. Необходимо обеспечить планомерную, совместную, скоординированную работу всех, кто заинтересован в решении данной задачи. Такое объединение возможно только при проведении единой государственной политики в области информатизации образования. Инструментом реализации этой политики на данном этапе образовательной реформы и должна стать данная Программа.

2. Цель и задачи Программы

Цель Программы — информационно-технологическое обеспечение реализации Закона Российской Федерации «Об образовании» и принципов государственной политики в области образования путем расширения применений средств и методов новых информационных технологий. Программа призвана координировать усилия органов государственной власти и управления образованием при разработке и реализации региональных и ведомственных Программ информатизации.

Для достижения этих целей необходимо решить следующие задачи:

- создание и внедрение необходимых учебно-методических материалов для реализации рекомендаций Базисного учебного плана в области информатизации;
- разработка и введение образовательных стандартов обучения информатике в образовательных учреждениях всех уровней;
- создание не менее 40 узлов образовательной информационной сети;
- создание при каждом узле сети банка учебно-методических материалов;
- создание условий вхождения российских образовательных учреждений в мировое образовательное информационное пространство;
- создание системы государственной сертификации аппаратных и программных средств информатизации образования;
- создание системы межрегиональных центров информатизации образования;
- координация региональных Программ развития образования.

3. Управление Программой

Реализация Программы проводится поэтапно под руководством Координационного комитета, состоящего из представителей федеральных и региональных органов управления, с оценкой результатов каждого этапа.

Все результаты, полученные в ходе выполнения Программы, являются общественным достоянием и предоставляются безвозмездно всем заинтересованным образовательным учреждениям нашей страны.

Финансирование Программы осуществляется по согласованным квотам из федерального, региональных и местных источников.

В качестве технической базы реализации проектов в рамках настоящей Программы используются современные средства ЭВТ и современные средства передачи информации, включая электронную почту.

Работа по информированию общественности о ходе выполнения Программы, а также распространение полученных результатов является составной частью самой Программы.

4. Основные разделы Программы (направления)

- I. Организационно—управленческое обеспечение Федеральной Программы развития образования (в области информатизации).
- II. Информатизация среднего образования.
- III. Информатизация подготовки и переподготовки педагогических и управленческих кадров.
- IV. Информатизация управления образованием.
- V. Обеспечение развития образования (в области информатизации).

I. Организационно-управленческое обеспечение Федеральной Программы развития образования (в области информатизации)*

1. Мониторинг реализации Программы информатизации образования в Российской Федерации (МО РФ, Ассоциация ИНТО).
2. Создание информационного образовательного агентства на базе корреспондентской сети журнала «Информатика и образование» (редакция журнала «Информатика и образование»).
3. Организация и проведение всероссийских семинаров, конференций, пленумов УМО по обмену информацией и опытом работы в области информатизации образования (МО РФ, Институт информатизации образования МО РФ).
4. Организация экспертизы и сертификации:
 - рукописей учебных и методических пособий, учебных программ;
 - программных средств учебного и управленческого назначения;
 - вычислительной техники и информационных технологий.

(Научно-методический центр МПГУ, Институт информатизации образования МО РФ, А/О КУДИЦ).

5. Создание Фонда педагогических программных средств учебного назначения и учебно-методических материалов, распространение информации о работе Фонда (Институт информатизации образования МО РФ, УМО ИПО).
6. Организация и проведение Всероссийского конкурса программных средств (Институт информатизации образования МО РФ).
7. Организация экспериментальных площадок и базовых образовательных учебных заведений по отработке моделей использования средств информатизации в учебном процессе (МО РФ).
8. Тиражирование и распространение результатов выполнения Программы по территориям России (редакция журнала «Информатика и образование»).

II. Информатизация среднего образования

1. Создание компьютерного курса «Информатика 5—6» (проблемная лаборатория Уральского ГПИ)
2. Создание компьютерного курса «Информатика 7—9» (предприятие ИнфоМир, проблемная лаборатория Уральского ГПИ).
3. Создание компьютерного курса «Информатика 10—11» для профильных школ (НИИ СИ РАН, проблемная лаборатория Уральского ГПИ).
4. Рекомендации по использованию отечественных и зарубежных педагогических программных средств в системе образования России (Институт информатизации образования МО РФ, ИПИ РАН, А/О «Прогресс»).
5. Создание полного учебно-методического комплекта компьютеризированного курса «Экология»
 - для 9 класса средней школы;
 - для 5 класса средней школы.

(проблемная лаборатория Уральского ГПИ, лаборатория телекоммуникации в образовании НСК РАН).

* Здесь приводятся направления работ и организации-исполнители по каждому из разделов.

6. Рекомендации по использованию средств мультимедиа в системе дополнительного образования (СП «ИНТЕРСОФТ»).
7. Разработка и утверждение стандартов содержания образования
 - по базовому курсу информатики;
 - по курсу информатики для профильных школ.
 (*МО РФ, Институт общеобразовательной школы РАО, Пермский ГПИ*).
8. Разработка и утверждение стандартов по ВТ, используемой в учебно-воспитательном процессе средних учебных заведений (*МО РФ, Ассоциация ИНТО, Институт новых технологий*).
9. Рекомендации по оборудованию и использованию кабинета информатики на базе средств НИТ (*Институт средств обучения РАО*).
10. Разработка санитарно-гигиенических норм использования различных типов компьютеров в учебно-воспитательном процессе образовательных учреждений, включая педагогические вузы (*Московский НИИ гигиены им. Ф. Ф. Эрисмана*).

III. Информатизация подготовки и переподготовки педагогических и управленческих кадров

1. Создание и внедрение учебно-методических комплексов по использованию НИТ в подготовке педагогических кадров (*УМО ИПО*).
2. Введение в педагогических вузах новых самостоятельных специальностей по информатике (*Уральский ГПИ, Тульский ГПИ, Омский ГПИ, Красноярский ГПИ*).
3. Разработка учебных программ и программно-методических комплексов по переподготовке педагогических кадров в системе региональных институтов повышения квалификации (*МИПКРО, Пермский ГПИ*).
4. Создание сети учебных центров по подготовке и переподготовке педагогических кадров в области использования новых информационных технологий в системе общего среднего образования России (*МО РФ*).
5. Организация и проведение курсов повышения квалификации управленческих кадров (*МО РФ, Нижегородский ГПИ, РИПКРО, Институт профессионального образования*).
6. Создание информационно-аналитической системы «Педагогическое образование» (*УМО ОППО*).
7. Компьютерно-ориентированные технологии подготовки профессионально-педагогических кадров для системы профессионально-педагогического образования (*Свердловский инженерно-педагогический институт*).

IV. Информатизация управления образованием

1. Создание телекоммуникационного узла Министерства образования Российской Федерации (*МО РФ*).
2. Создание и эксплуатация информационной базы «Образование» с выходом на международные образовательные сети (*МО РФ, РИПКРО (РИПЦ)*).
3. Создание и эксплуатация информационной базы по образовательным стандартам и программам в системе начального профессионального образования (*Институт профессионального образования*).
4. Создание и эксплуатация образовательной телекоммуникационной сети органов управления образованием (*МО РФ*).
5. Создание нормативно-методической документации по эксплуатации образовательной телекоммуникационной сети (*Институт информатизации образования МО РФ, Рязанский ГПИ*).

6. Создание системы управления для уровней:

- федерального;
- регионального;
- районного;
- образовательного учреждения.

(Ассоциация «Компьютер и детство», Институт проблем передачи информации РАН, Московский технологический университет «СТАНКИН»).

V. Обеспечение развития образования (в области информатизации)

1. Разработка Программы информатизации образования и рекомендаций к разработке региональных программ информатизации образования на 1996—1998 гг. и до 2000 г (*МО РФ, Институт проблем передачи информации РАН, Институт общеобразовательной школы РАО*).
2. Разработка концепции и программы «НИТ в обучении детей-инвалидов» (*Институт коррекционной педагогики РАО*).
3. Создание компьютерных курсов для учащихся 10—11 классов в системе дополнительного образования (*предприятие ИнфоМир, Уральский ГПИ, Институт новых технологий*).
4. Создание программного комплекса для работы на основе иерархической информационной технологии и образных представлений (*лаборатория визуальных систем Института математики и механики Уральского отделения РАН*).
5. Создание мультимедиа-курсов по различным образовательным дисциплинам (*А/О КУДИЦ, Институт средств обучения РАО, МИПКРО*).
6. Создание учебной программы сквозного курса «Информационная культура» для 1—11 классов (*ВНТК, г.Переславль-Залесский*).
7. Создание учебно-методической модели непрерывного обучения информатике в системе «Школа—педагогический вуз» (*РГПУ им. А. И. Герцена, г. Санкт-Петербург*).
8. Рекомендации по использованию средств малой электроники в системе дополнительного образования (*Учебно-методический центр «ЭЛТИ»*).
9. Рекомендации по использованию компьютерных тренажеров для умственного и физического развития дошкольников с учетом медицинского контроля и специфики возраста (*Ассоциация «Компьютер и детство»*).
10. Организация и проведение Всероссийских и международных олимпиад школьников (*лицей № 542, г. Москва*).
11. Организация и проведение региональных летних компьютерных школ (*Центр информационных технологий, г. Миасс; Ассоциация учителей информатики, г. Красноярск*).

Приложение
к «Программе информатизации образования
в Российской Федерации на 1994—1995 гг.»

ОСНОВНЫЕ НАПРАВЛЕНИЯ РЕГИОНАЛЬНОЙ ПРОГРАММЫ ИНФОРМАТИЗАЦИИ

1. Введение

Начавшийся в 1985 г. процесс компьютеризации учебных заведений и информатизации образования соответствовал объективным интересам развития общества — экономики, науки и техники, образования и культуры. Политические и социально-экономические изменения, произошедшие в Российской Федерации в последнее время, сделали задачу информатизации общества и, как необходимое условие этого, информатизация образования еще более актуальной.

Глобальность поставленных задач, необходимость приобщения к современным информационным технологиям значительной части общества подразумевает начало их решения на уровне среднего звена образования. Сегодня во многих странах выпускник средней школы либо сразу, либо после относительно небольшого дополнительного профессионального обучения приходит на рабочее место, оснащенное компьютером или средствами связи, делопроизводства на базе микропроцессорной техники, и должен для успешной работы представлять себе принципы ее функционирования.

Вместе с тем даже получение всеми членами общества базовой подготовки в области информатизации в средней школе не может сделать российское общество действительно информационным. Создание и развитие информационной инфраструктуры, без которой невозможно обеспечение доступности информации всем членам общества, необходимость широкого внедрения НИТ во всех отраслях народного хозяйства предъявляют новые требования к профессиональной подготовке кадров. Таким образом, информатизация профессионального образования становится одним из важнейших направлений информатизации образования.

Информатизация образования невозможна без соответствующей подготовки и переподготовки педагогических кадров. Освоение НИТ учащимися возможно только при опережающей подготовке преподавателей в этой области. Поэтому информатизация педагогического образования, подготовка учителей по новым специальностям, освоение всеми преподавателями информационных технологий являются приоритетным, определяющим направлением для всей информатизации образования.

Наконец, чрезвычайно важной является реализация потенциальной эффективности информатизации в самой сфере образования. Мировой и отечественный опыт указывают на возможность существенно изменить к лучшему содержание и формы учебного процесса, внедрить в образование новые информационные технологии, что способствует решению задач интенсификации и индивидуализации обучения, формированию познавательных способностей учащихся, улучшению организации учебного процесса, распространению передового педагогического опыта, реальному обеспечению права всех граждан на получение полноценного образования.

2. Необходимость региональной Программы информатизации

Реализация информатизации образования в период 1985-1992 гг. отражала всю политико-экономическую нестабильность общества. С одной стороны, в нее были вложены значительные средства, выпущены сотни тысяч школьных персональных ЭВМ, созданы тысячи специально ориентированных компьютерных программ, изданы учебники и учебные пособия. С другой стороны, уровень и качество этих компьютеров низки, их многотипность и несовместимость между собой ошеломляют, качество подавляющего большинства педагогических программных средств ниже всякой критики.

Общегосударственные программы 1985-1990 гг. не только не решали проблемы оснащения образовательных учреждений вычислительной техникой, но и не выработали реальных критериев ее отбора.

Так и не был создан централизованный фонд программного обеспечения и учебно-методических материалов.

В таких условиях сложились резко различные региональные варианты реализации компьютеризации среднего образования. Пользуясь неразберихой и отсутствием координации, недобросовестные производители технических и программных средств заваливают народное образование заведомо некачественной продукцией. Так, в ряде школ появились совершенно непригодные к работе, сделанные кустарно компьютерные классы; директора, не разбираясь в этих вопросах, покупают на многие тысячи рублей ненужные, выполненные на низком уровне программы.

Возросшая в последнее время роль регионов в экономической, научно-технической жизни страны не позволяет им не рассматривать вопросы развития образования. Разрабатываются региональные стандарты образования, администрации на местах уделяют ему разные доли бюджета и разную степень внимания. Нет смысла надеяться, что все проблемы информатизации образования могут быть решены за счет принятия федеральных программ. Опыт прошедших лет показывает, что реальные условия и потребности регионов не знает никто лучше их самих.

Финансирование большинства образовательных учреждений осуществляется из регионального бюджета, и эти средства должны расходоваться с максимальной пользой для самого региона. В таких условиях кивать на отсутствие единой всеохватывающей государственной программы в вопросах компьютеризации и информатизации образования, образно говоря, все равно что заказывать синоптикам погоду.

3. Цели и задачи региональной Программы информатизации

Целью информатизации образования территорий является существенное повышение качества образования за счет наиболее полного удовлетворения образовательных потребностей населения (в том числе и в овладении информационными технологиями); повышения информированности как внутри самой системы образования, так и населения региона; повышения уровня компетентности управления образованием.

Для достижения этой цели необходимо создание Программы, предусматривающей решение следующих задач:

1. Осуществление предпроектного обследования системы образования территории для определения:

- сложившейся структуры компьютерного и программного оснащения учебных заведений;
- объективных потребностей региона в целом и сферы образования в частности;
- материальных ресурсов, которые территория может в ближайшие годы предоставить для компьютеризации и информатизации образования.

2. Создание системы организационного и научно-методического обеспечения программы информатизации, охватывающей следующие направления:

- координацию всех работ по информатизации образования внутри региона, а также совместное выполнение работ с другими регионами и в рамках Федеральной Программы информатизации;
- контроль за соблюдением государственных стандартов образования в области информатизации;
- формирование рынка программных и аппаратных средств информатизации, учебно-методических материалов, других образовательных товаров и услуг;
- подготовку и переподготовку кадров;
- обеспечение всех видов методической работы;
- создание экспериментальных площадок;
- создание регионального банка педагогических программных средств и учебно-методических материалов;
- создание региональной информационной инфраструктуры на основе средств телекоммуникации;
- сбор и обработку педагогической информации, ее доведение до учреждений образования;
- приобретение техники, ее установку, текущее обслуживание, ремонт.

Решение этих задач может быть возложено на один или несколько специальных центров, создаваемых в рамках региональной Программы информатизации.

4. Структура региональной Программы информатизации образования

Министерство образования Российской Федерации предлагает регионам страны в качестве рекомендации следующую структуру региональной Программы информатизации образования.

I. Объяснительная записка (мотивация необходимости Программы).

II. Анализ существующего в регионе положения в области компьютеризации и информатизации образования.

1. Техническое оснащение:

- а) виды КУВТ;
- б) техническое состояние, реальная работоспособность;
- в) организация ремонта и обслуживания.

2. Наличие подготовленных кадров и качество подготовки:

- а) учителей информатики;
- б) учителей-предметников;
- в) служащих и руководителей органов управления образованием;
- г) других категорий работников образования.

3. Наличие программного обеспечения по каждому из видов КУВТ, его доступность и качество:

- а) прикладных программ общего назначения;
- б) инструментальных программных средств;
- в) программной поддержки курса информатики;
- г) программной поддержки других дисциплин.

4. Учебно-методическое оснащение, его достаточность:

- а) учебники, пособия, изданные централизованно;
- б) региональные разработки.

III. Основные направления развития информатизации образования региона на _____ гг.

1. Научно-методическое обеспечение учебно-воспитательного процесса (учебно-методическая литература, педагогические программные средства и т. п.):

- а) разработка собственными силами;
- б) создание на основе кооперации с другими регионами;
- в) использование результатов Федеральной Программы (в области информатизации).

2. Техническое оснащение:

- а) определение технической политики и источников приобретения аппаратных средств информатизации;
- б) приобретение и установка;
- в) организация ремонта и обслуживания.

3. Подготовка и переподготовка кадров:

- а) осуществление собственными силами;
- б) необходимость кооперации с другими регионами;
- в) участие в семинарах и курсах, организуемых федеральными органами управления образованием.

4. Программное обеспечение управления образованием:

- а) разработка собственными силами;
- б) создание на основе кооперации с другими регионами;
- в) использование результатов Федеральной Программы (в области информатизации).

5. Развитие региональной телекоммуникационной сети образования, подключение к общей сети образования.

6. Организационное обеспечение.

7. Финансовое обеспечение.

IV. Совместные проекты в области информатизации образования с другими регионами и международное сотрудничество.

V. Ожидаемые результаты и механизмы контроля.

Приложение 1
к «Основным направлениям региональной
Программы информатизации»

**ПРИМЕРНОЕ ПОЛОЖЕНИЕ
О РЕГИОНАЛЬНОЙ АССОЦИАЦИИ
ИНФОРМАТИЗАЦИИ ОБРАЗОВАНИЯ**
(название координирующей организации условное)

1. Общие положения

Ассоциация информатизации образования — общественное формирование, объединяющее в своем составе на добровольной основе всех заинтересованных в решении задач информатизации образования юридических и физических лиц.

Ассоциация не имеет статуса юридического лица, ее деятельность не носит коммерческого характера.

Структура Ассоциации и направления деятельности ее подразделений определяются региональной Программой информатизации образования, разработанной по заказу органов управления региона и одобренной органами управления образованием региона.

В рамках Ассоциации могут создаваться любые подразделения и структуры, не противоречащие в своей деятельности законодательству Российской Федерации и согласующиеся в направлениях деятельности с Ассоциацией. Эти подразделения могут быть юридическими лицами и пользоваться полной экономической самостоятельностью.

2. Цели и задачи Ассоциации информатизации образования

Основная цель Ассоциации — разработка и обеспечение совместно с заинтересованными лицами реализации комплексной программы информатизации образования.

Задачи Ассоциации:

- обеспечение единого научно-методического руководства процессом информатизации образования;
- выработка и обеспечение единой политики в процессах информатизации образования;
- кооперация всех заинтересованных в процессах информатизации образования сил;
- координация процессов обеспечения учреждений образования вычислительной техникой и педагогическими программными средствами;
- научно-методическое сопровождение внедрения новых информационных технологий в учебно-воспитательный процесс;
- научно-методическое сопровождение процессов подготовки педагогических кадров к использованию ЭВТ;
- обеспечение создания и наполнения региональных банков педагогической и программной информации;
- создание единого регионального информационного пространства системы образования региона и соответствующих телекоммуникаций;
- оказание практической помощи подведомственным учреждениям в вопросах технического оснащения и обслуживания ЭВТ и оргтехники.

3. Руководство Ассоциацией

Общее руководство Ассоциацией осуществляет Совет Ассоциации, в состав которого входят по одному представителю от каждого подразделения. Сопредседателями Совета Ассоциации являются научный руководитель и координатор. Заседания Совета проводятся не реже одного раза в 2 месяца. Общее собрание сотрудников всех структурных подразделений Ассоциации проводится не реже 1 раза в год.

4. Структура Ассоциации (возможный состав членов Ассоциации)

1. Региональный информационно-методический Центр при региональном институте повышения квалификации работников образования.
2. Региональный центр подготовки педагогических кадров.
3. Предприятие по ремонту учебной вычислительной техники
4. Центр информационных технологий профессионально-технического образования.
5. Экспертный совет.

Приложение 2
к «Основным направлениям региональной
Программы информатизации»

ПРИМЕРНОЕ ПОЛОЖЕНИЕ О РЕГИОНАЛЬНОМ ИНФОРМАЦИОННО-МЕТОДИЧЕСКОМ ЦЕНТРЕ

1. Общие положения

Региональный информационно-методический центр (далее — Центр) является структурным подразделением регионального ИПКРО, осуществляет свою деятельность в соответствии с Основными направлениями развития образования региона, координирует ее с Ассоциацией информатизации образования.

В своей практической деятельности Центр руководствуется законодательными и нормативными актами Министерства образования Российской Федерации, РИПКРО, региональных органов управления образованием, Уставом регионального ИПКРО и настоящим Положением.

Центр создан для осуществления научно-методического обеспечения процессов выявления, изучения и пропаганды передового педагогического опыта и инноваций в регионе образования, поэтапного решения задач информатизации образования, внедрения новых информационных технологий в образовательный и управленческий процессы, создания областного банка педагогической информации, создания, развертывания и обслуживания региональной сети модемной и факсимильной связи учреждений народного образования в регионе.

Центр создает, накапливает и систематизирует педагогическую информацию, представленную в текстовом, графическом, аудиовизуальном и иных видах.

Центр производит поставку информационных материалов в учреждения образования, педагогам области в любой форме: видео- и кинофотоматериалы, файлы на дискетах для ПЭВМ, посредством модемной и факсимильной связи, в виде печатной продукции.

2. Задачи Центра:

- сбор, накопление, обработка, систематизация, обобщение и распространение педагогической информации в соответствии с принятыми в России стандартами в системе образования;
- выявление информационных потребностей и удовлетворение запросов педагогических кадров региона по новым информационным технологиям и педагогическим инновациям;
- координация работы кафедр, подразделений ИПК и районных информационно-методических кабинетов в информационно-аналитической деятельности;
- формирование массива педагогической информации, работа с ним и включение региональной педагогической информации в республиканскую информационную сеть после научно-методической экспертизы кафедрами ИПК и районными информационно-методическими кабинетами;

- осуществление взаимодействия с информационными центрами и службами республиканского и регионального уровней для расширения банка педагогической информации, развития и координации информационных потоков;
- осуществление посреднических услуг по удовлетворению запросов пользователей (органы управления образованием, образовательные учреждения, педагоги, родители, ученики), доставке информации о достижениях психолого-педагогической науки, о новых педагогических и информационных технологиях;
- организация повышения квалификации работников образования по методике формирования, изучения, обобщения и распространения педагогического опыта и педагогической информации;
- поэтапное создание региональной медиатеки, фонда педагогических программных средств и аудиовизуальных средств обучения.

3. Направления деятельности Центра:

- создание банка педагогической информации как основы единой региональной информационной сети;
- разработка программного обеспечения информационно педагогических модулей на различных носителях (дискеты, видео- и магнитные записи);
- разработка системы сбора педагогической информации из различных источников, региональной системы компьютерной обработки педагогической информации и ее распространение в сети электронных средств;
- оказание маркетинговых и консалтинговых услуг потребителям педагогической информации;
- анализ и экспертиза педагогической информации, организация информационных потоков;
- разработка методических материалов для сбора педагогической информации, изучение и обобщение ППС и организация деятельности информационно-методических кабинетов;
- организация распространения педагогической информации через издательскую деятельность, аудиовизуальные программы, электронную почту;
- подготовка пользователей единой региональной информационной системы;
- оказание методической и консультационной помощи сотрудникам подразделений ИПК и различным категориям педагогических кадров в получении информации от Центра и предоставлении информации в его банк;
- создание компьютерных систем и систем кабельного учебного телевидения в интересах образования и сопутствующих сфер;
- выполнение работ, связанных с техническим перевооружением учреждений образования; разработка, создание, обслуживание и реализация научно-технической продукции широкого назначения;
- другие виды деятельности, соответствующие профилю Центра, не запрещенные законодательством.

4. Структура и штаты

Структура и штаты Центра, условия, и порядок оплаты труда определяются в соответствии с действующим законодательством, Уставом регионального ИПКРО и инструкциями.

В структуру Центра входят:

- отдел программно-методического обеспечения регионального банка педагогической информации;
- отдел аудиовизуальных средств обучения и медиатека;
- редакционно-издательский отдел;
- отдел сбора и обработки инновационной педагогической информации.

5. Финансовая и хозяйственная деятельность

5.1. Финансирование Центра происходит из двух основных источников:

- а) основные средства бюджета — сотрудники Центра являются штатными сотрудниками Регионального ИПКРО (согласно штатному расписанию).
- б) за счет поступлений от хозяйственной деятельности Центра. Создаваемые временные трудовые коллективы, для выполнения договорных работ, оплачиваются из фондов заработной платы по договору.

5.2. Центр дополнительно осуществляет хозрасчетные работы, не являющиеся приоритетными в деятельности Центра, — прямые договоры с учреждениями народного образования, иными организациями, частными лицами на разработку, тиражирование программных продуктов, аудио-видео-материалов, пусконаладочные и ремонтно-восстановительные работы средств ВТ и видео-техники.

5.3. Приобретение техники, разработка и адаптация программных средств, видеоматериалов происходит за счет целевых средств регионального УНО и средств, полученных от хозрасчетной деятельности.

Приложение 3
к «Основным направлениям региональной
Программы информатизации»

ПРИМЕРНОЕ ПОЛОЖЕНИЕ О РЕГИОНАЛЬНОМ ЦЕНТРЕ ПОДГОТОВКИ ПЕДАГОГИЧЕСКИХ КАДРОВ

1. Общие положения

1.1. Региональный Центр подготовки педагогических кадров создается на основании Учредительного договора между юридическими лицами, а именно:

- Региональный орган управления образованием;
- Педагогический институт (университет);
- Институт повышения квалификации работников образования (ИПКРО).

1.2. Центр действует на основании Закона Российской Федерации «Об образовании», другого действующего Российского законодательства, а также Устава и Учредительного договора.

Центр реализует в своей деятельности федеральную и региональную политику в вопросах компьютерной подготовки педагогических кадров и информатизации образования региона, обеспечивает курсовую подготовку всех категорий педагогов и управленческого персонала подведомственных учреждений к работе на ЭВМ, совместно с ПИ и ИПК создает региональный банк педагогических программных средств для всех типов ЭВМ, используемых в учреждениях образования региона.

1.3. Центр по своему статусу является государственным образовательным учреждением, реализующим дополнительную образовательную программу и услуги.

1.4. Центр с момента регистрации является юридическим лицом, обладает обособленным имуществом, имеет самостоятельный баланс, расчетный и иные счета в российских и зарубежных банках, эмблему и печать со своим наименованием.

2. Цели, задачи, функции Центра

2.1. Целью создания Центра является реализация дополнительного образования и услуг для всестороннего удовлетворения потребностей граждан, общества, государства в области информатизации и компьютеризации образования региона.

2.2. Основной задачей Центра является непрерывное повышение квалификации учителей, служащих и специалистов сферы образования в связи с постоянным повышением образовательных стандартов в области информатизации и компьютеризации.

2.3. Для выполнения своих целей и задач Центр осуществляет следующие виды деятельности:

- организует и проводит учебные процессы по подготовке и переподготовке кадров в области информатизации и компьютеризации образования;
- выполняет работы, связанные с техническим перевооружением учреждений образования;
- разрабатывает, создает, обслуживает и реализует научно-техническую продукцию широкого назначения, в том числе: автоматизированные системы управления образованием,

банки данных, информационные фонды, пакеты прикладных программ, автоматизированные рабочие места, средства связи, оргтехники, лабораторное оборудование;

- разрабатывает и внедряет программное обеспечение в системе образования;
- проводит исследования, экспертизы, разработки педагогических программных средств для всех типов ЭВМ, применяемых в учреждениях образования региона, создает региональный стандарт образования по информатике и вычислительной технике и по компьютерным компонентам подготовки по различным предметам;
- распространяет опыт компьютеризации и информатизации в учебных учреждениях региона;
- организует и проводит конференции, семинары, симпозиумы, деловые встречи, бизнес-туры, круизы как в РФ, так и за рубежом;
- оказывает различные услуги в области информатизации и компьютеризации;
- разрабатывает, издает и тиражирует научно-методическую литературу, видео- и аудио-продукцию по проблеме информатизации и компьютеризации;
- проводит маркетинг товаров и услуг, обеспечивающих компьютеризацию и информатизацию;
- рекламирует товары и услуги в указанной области.

2.4. Деятельность Центра, связанная с образовательным процессом, осуществляется на основании дополнительной образовательной программы, принимаемой в установленном Законом порядке. В программе предусматриваются сроки ее освоения, организация учебных процессов, порядок приема обучающихся, форма и порядок аттестации слушателей и другие вопросы, регламентирующие образовательный процесс.

2.5. Право на образовательную деятельность и льготы, предоставляемые законодательством Российской Федерации, возникают у Центра с момента выдачи ему лицензии (разрешения).

3. Руководство, финансирование

3.1. Центр возглавляют директор, принимаемый на контрактной основе по взаимному согласию учредителей, и научный руководитель. Директор и научный руководитель имеют право подписи банковских документов.

Структура центра, штатное расписание утверждаются начальником регионального органа управления образованием.

3.2. Прием на работу сотрудников Центра осуществляет директор по согласованию с научным руководителем.

3.3. Финансирование Центра осуществляется региональным органом управления образованием согласно смете, утверждаемой ежегодно. Центр имеет право хозрасчетной деятельности, прибыль от хозрасчетной деятельности распределяется по взаимному согласию учредителей.

3.4. Организация работы Центра определяется Уставом; права и обязанности учредителей определяются Учредительным договором.

Приложение 4

к «Основным направлениям региональной Программы информатизации»

О ПРЕДПРИЯТИИ ПО РЕМОНТУ УЧЕБНОЙ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Основными направлениями деятельности предприятия являются:

- осуществление работ по техническому обслуживанию (ТО) учебной вычислительной техники в учреждениях образования. Работа проводится по договорам, заключенным на год, с дальнейшим их продлением. Оплата производится по актам выполненных работ поквартально или по счетам предварительно. При техническом обслуживании обеспечивается полная исправность класса ВТ;

- ввод в эксплуатацию новых компьютерных классов с выполнением всех электромонтажных работ и сдачей полностью исправного оборудования;
- поставка с заводов-изготовителей компьютерных классов отечественного производства, рабочих мест преподавателя, отдельных учебных мест, отдельных устройств по всей номенклатуре, выпускаемой заводами;
- сотрудничество с другими предприятиями, осуществляющими дополнительную модернизацию, разработку новых устройств, позволяющих улучшить характеристики школьной ВТ;
- подготовка представителей предприятий на местах, оснащение их необходимым оборудованием, обменным фондом, обучение их для более оперативного и качественного обслуживания школ на местах в районах области;
- создание и совершенствование производственной базы (приобретение новых стендов, дефицитного ЗИПа, основных фондов);
- обеспечение ремонта оргтехники и иного электронного оборудования, эксплуатируемого в учреждениях образования области.

Приложение 5
к «Основным направлениям региональной Программы информатизации»

О ЦЕНТРЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ НАЧАЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

1. Центр информационных технологий начального профессионального образования (далее — Центр) создается региональным органом управления образованием для координации работ по внедрению новых информационных технологий обучения в профессиональных учебных заведениях региона.

2. На Центр возлагается выполнение следующих работ:

2.1. Изучение состояния дел с внедрением новых информационных технологий в профессиональных учебных заведениях:

- сбор информации о наличии вычислительной техники (ВТ) и программно-педагогических средств (ППС);
- выяснение проблем с преподаванием информатики и использованием ВТ;
- изучение передового опыта.

2.2. Приобретение ППС и методических разработок:

- связь с организациями — разработчиками ППС по специальным и общетехническим дисциплинам;
- ознакомление с ППС и методическими материалами, их приобретение и подготовка к передаче профессиональным учебным заведениям.

2.3. Разработка ППС:

- выбор перечня ППС для первоочередной разработки; привлечение опытных преподавателей и мастеров п/о к разработке сценариев ППС;
- организация разработки и экспертизы ППС;
- адаптация ППС на другие типы ВТ;
- подготовка разработанных ППС к передаче профессиональным учебным заведениям.

2.4. Тиражирование и передача ППС по заявкам профессиональных учебных заведений.

2.5. Оказание профессиональным учебным заведениям помощи во внедрении ППС в учебный процесс.

2.6. Приобретение, разработка и поставка по заявкам профессиональных учебных заведений программно-технических комплексов (ПТК):

- сбор заявок;
- изучение конъюнктуры рынка;
- выполнение заявок, передача ПТК заказчикам.

2.7. Монтаж, пусконаладка и обслуживание техники (выполняется по заявкам профессиональных учебных заведений). Обучение персонала.

2.8. Автоматизация функций управления в рамках «АСУ — ПРОФЕССИОНАЛЬНОЕ ЗАВЕДЕНИЕ».

2.9. Внедрение в учебно-воспитательный процесс профессиональных учебных заведений видеотехники и телевидения.

2.10. Информационное обслуживание преподавателей информатики и других пользователей ВТ из профессиональных учебных заведений:

- выпуск каталогов ППС;
- подготовка материалов по обмену опытом;
- периодическая рассылка информационных и рекламных материалов.

2.11. Обучение, стажировка и консультирование работников профессиональных учебных заведений, организация обмена опытом:

- сбор заявок и организация различных видов обучения;
- обеспечение постоянно действующей системы консультаций;
- проведение семинаров, олимпиад и т. п.

2.12. Разработка и изготовление наглядных пособий, раздаточных и методических материалов.

3. Центр привлекает к выполнению работ на договорной основе сотрудников профессиональных учебных заведений, а также сторонние организации и специалистов.

4. Работа Центра с профессиональными учебными заведениями ведется на основе договоров. В случае централизованного финансирования работ по приобретению и разработке ППС. Центр передает их профессиональным учебным заведениям бесplatно.

Приложение 6
к «Основным направлениям региональной
Программы информатизации»

ОБ ЭКСПЕРТНОМ СОВЕТЕ

*(Положение и состав утверждаются
региональным органом управления образованием)*

Задачи Экспертного совета:

- своевременная и качественная экспертиза педагогических программных средств, приобретаемых для создания регионального банка программных средств;
- экспертиза учебных планов и программ средних учебных заведений по основам информатики и вычислительной техники и другим предметам, содержащим элементы информационных технологий и использования ЭВТ;
- определение перспективных моделей технического перевооружения системы образования;
- выдача заключений по вопросам целесообразности вложения ассигнований в разработку, адаптацию технических и программных средств.

И_нФ_оМ_иР

предлагает:

- КуМир-93 (КуМир-Гипертекст) – НОВАЯ ВЕРСИЯ для IBM PC.*
- первые 10 заявок – по цене UPGRADE (40% от цены по каталогу) !*
- Функции и Графики, Планиметрия, Информатика-10 – на базе КуМира-93*
- МикроМир, ФортранМир, ПаскальМир для IBM PC*
- КуМир, ФортранМир для УКНЦ*
- КорветМир, ЯмахаМир*

**Вниманию руководителей органов управления образованием!
При централизованных закупках предоставляются значительные скидки!**

Напоминаем зарегистрированным пользователям о необходимости направлять в «ИнфоМир» регистрационные карточки. Летом 1994 года будет проведена лотерея среди зарегистрированных пользователей.

NEW !!!

МУЛЬТИМЕДИА-КуМир

NEW !!!

Мультимедиа-КуМир — это:

- все возможности системы КуМир (полная поддержка школьного алгоритмического языка, описанного в учебнике А.Г. Кушниренко и др.; мгновенная и наглядная диагностика всех ошибок ученика при вводе и редактировании программы; встроенный отладчик и т.д.);*
- возможность подготовки гипертекстов для уроков информатики и математики с использованием мультимедиа-технологии (совмещение текста, графики, звука и мультимедиа), не требующая программирования на специальных языках;*
- практикумы с возможностью получения подсказки по задаче и проверки ответа;*
- проведение автоматизированных контрольных работ;*
- готовый комплект гипертекстов для каждого урока информатики, обучающие гипертексты по школьному алгоритмическому языку и системе КуМир, включая исполнители Робот, Чертежник, Вездеход, Куберия, Гратекс, Жизнь;*
- работа в локальной сети, возможности производственного применения и многое другое.*

Заявки направляйте по адресу:

103051, Москва, Садовая Сухаревская, 16, комн.9.

Редакция журнала «Информатика и образование». Отдел рекламы.

В. К. Белошапка,
д-р физ.-мат. наук,
А. С. Лесневский,
канд. пед. наук

ТРЕБОВАНИЯ К ЗНАНИЯМ И УМЕНИЯМ ШКОЛЬНИКОВ ПО ИНФОРМАТИКЕ

Информатика как наука, а тем более как учебный предмет находится в стадии становления. Существуют различные точки зрения, что такое информатика и как следует преподавать



Рис. 1

данный предмет в школе. Изложению нашей точки зрения на это посвящена серия публикаций [1] — [7]. В данной статье вниманию читателя предлагается система требований к знаниям и умениям по информатике, логически из них вытекающая. Эти требования могут служить для разработки стандартов по базовому курсу информатики.

Схема на рис. 1 демонстрирует три аспекта процесса освоения и познания мира человеком [2]. Отражение мира в соответствующей понятийной среде, т. е. собственно познание, — это прерогатива ученого, использование научных знаний для создания технологий — инженера, использование готовых технологий — прерогатива рабочего.

Эта схема может быть применена и к анализу содержания образования, в частности курса информатики (рис. 2).



Рис. 2

Здесь каждому элементу соответствует определенный аспект:

«буквы» — формальные исчисления и алгоритмы,

«компьютер» — устройство компьютера,

«мир» — социально-мировоззренческий аспект.

Каждое бинарное отношение также порождает свою линию:

«мир — буквы» — информационное моделирование,

«буквы — компьютер» — программирование,

«компьютер — мир» — пользовательский аспект.

Решение задач с помощью компьютера носит интегративный характер по отношению к вышеперечисленным.

Комплекс научно-инженерных знаний по информатике включает три составляющие:

- аппаратное обеспечение (hardware),
- программное обеспечение (software),
- интеллектуальное обеспечение (brainware).

Условно рассматривая два классификационных ряда: «пользователь - программист — ученый» и «hard — soft — brain», мы получим таблицу, которая позволяет дать более подробный тематический обзор содержания учебного предмета. При этом мы не будем ограничиваться рамками средней школы.

Таблица

	Пользователь	Программист	Ученый
Hard	Начальные сведения об устройстве компьютера и техника безопасного обращения с ним	Принципы устройства и архитектура компьютера	Физические принципы построения компьютерных систем
Soft	Типовое программное обеспечение, его назначение, использование	Технология программирования	Теория построения программных систем
Brain	Возможности и ограничения компьютера и его место в современном мире, представления о моделях и алгоритмах	Теория алгоритмов, теоретическое программирование	Формальные исчисления и информационное моделирование

В средних классах, предположительно в 6-ом, необходим небольшой вводно-ознакомительный курс, имеющий «пользовательскую» ориентацию. Это позволит эффективно использовать компьютер в преподавании других предметов. Одновременно в курсе математики можно познакомить учащихся с основами формального исчисления и алгоритмизации, а в курсе физики — с принципами устройства и работы компьютера.

В старших классах желательно подходить к продолжению обучения информатике дифференцированно. Нужно формировать специализированные курсы: курс, ориентированный на продолжение подготовки пользователя и являющийся углубленным вариантом вводно-ознакомительного; курс с акцентом на принципы и технологию программирования; курс, ориентированный на теоретические принципы и практические навыки построения информационных моделей и работы с ними. В соответствии с этим дифференцированы и требования. Они построены по содержательно-методическим линиям, которые возникают из рассмотрения трех

планов бытия и их взаимодействия применительно к информатике: «мир — человек — знаковая среда».

Первая линия основана на изучении свойств собственно знаковой среды: формальное исчисление и алгоритмы (I).

Из рассмотрения того, как мир отражается в знаковой среде, возникает линия информационного моделирования (II).

Освоение человеком знаковой среды порождает инструментально-технологическую линию (III), которая имеет следующие аспекты: инструментально-аппаратный (компьютер), инструментально-программный (программное обеспечение), собственно технологический (технология решения задач с использованием компьютера). Сюда же естественно примыкает комплекс социально-мировоззренческих проблем (феномен информатизации).

Ввиду многообразия существующих методических и содержательных подходов к преподаванию информатики в школе требования отражают целевые установки по отношению к итоговым результатам обучения на старшей ступени школы независимо от того, в какой ступени будет изучаться тот или иной материал. При этом предполагается, что предмет преподается в школах, оснащенных компьютерами.

6—7 классы

Формальное исчисление и алгоритмы (I)

Изучение материала дает возможность учащимся:

- понимать смысл требования «составить алгоритм решения задачи»;
- знать основные алгоритмические конструкции (цикл, ветвление, серия, процедура) и уметь использовать их для записи простейших алгоритмов в виде блок-схем;
- иметь представление о понятиях: данные, структура данных, входные данные (аргументы), выходные данные (результаты).

Уровень обязательной подготовки определяется следующими требованиями:

- уметь составлять последовательность точно сформулированных предписаний на естественном языке, описывающую ход решения какой-либо задачи;
- уметь представлять ход решения задачи в формализованном виде;
- уметь определять входные данные (дано) и выходные данные (требуется) для задачи.

Инструментально-технологическая линия (III)

Здесь необходимо:

- знать основные сферы применения компьютеров;
- знать основные устройства персонального компьютера и их назначение;
- иметь представление об устройстве компьютера.

10—11 классы

Формальное исчисление и алгоритмы (I)

Обучение в школе дает возможность ученику:

- овладеть понятиями и аппаратом исчисления высказываний: «истина», «ложь», высказывания, связки, кванторы;
- понимать соотношение естественного языка и формального исчисления;
- понимать смысл понятия «алгоритм»;
- усвоить основные алгоритмические конструкции (цикл с предусловием, цикл с постусловием, цикл с параметром, серия, ветвление, выбор, процедура) и типы данных (числовые и текстовые величины, массив, связанный список, дерево) и научиться использовать их для решения задач на составление алгоритмов;
- научиться использовать какой-либо язык программирования для записи алгоритмов;

- овладеть методами составления алгоритмов;
- научиться определять правильность алгоритма по его записи;
- ознакомиться с современными тенденциями в технологии программирования.

Уровень обязательной подготовки определяется следующими требованиями:

- знать содержание понятий: знак, буква, информация;
- знать основные типы логических связей;
- уметь составлять таблицы истинности для простейших логических формул;
- уметь записывать предложения на естественном языке в виде формул исчисления высказываний;
- понимать смысл требования «составить алгоритм решения какой-либо задачи»;
- решать стандартные типы задач на составление алгоритмов;
- распознавать основные алгоритмические конструкции в записи алгоритма на одном из языков программирования.

Информационное моделирование (II)

Изучение материала дает возможность учащимся:

- получить представление о моделировании как о методе познания и об основных типах моделирования (математическое, физическое и пр.);
- понимать смысл отношений «объект — модель — язык»;
- знать смысл аналогии «язык — средства моделирования»;
- получить представление о методе информационного моделирования как методе современного естествознания;
- овладеть навыками структурирования, т. е. уметь строить описания объектов в терминах «элемент», «свойство», «отношение»;
- научиться использовать графы для представления простейших моделей;
- знать основные виды моделей (классификационные, динамические и формально-языковые) и сферы их применимости;
- понимать связь типов программных средств и видов информационных моделей;
- знать сущность метода информационного моделирования.

Уровень обязательной подготовки определяется следующим требованием:

- знать место информационного моделирования в основной технологической цепочке.

Инструментально-технологическая линия (III)

Компьютер

Изучение материала дает возможность учащимся:

- прийти к пониманию того, что компьютер — это универсальное орудие для работы с формально-знаковыми конструкциями;
- ознакомиться с устройством современных компьютеров;
- понять принципы функционирования компьютера;
- понять принципы кодирования и представления информации в компьютере;
- ознакомиться со спектром современной компьютерной техники, ее характеристиками и областями применимости.

Уровень обязательной подготовки определяется следующими требованиями:

- знать функциональную схему организации компьютера;
- знать принцип автоматического исполнения программы;
- иметь представление о характеристиках основных типов современных компьютеров;
- иметь представление о принципах кодирования информации;
- знать технику безопасности при работе с персональным компьютером.

Программное обеспечение

Изучение материала дает возможность учащимся:

- ознакомиться с основными типами программного обеспечения и научиться ими пользоваться (текстовый редактор, электронные таблицы, база данных, графический редактор, экспертная система);
- понимать назначение и принципы работы основных типов программного обеспечения.

Уровень обязательной подготовки определяется следующими требованиями:

- знать названия и назначение основных типов программного обеспечения;
- уметь пользоваться типовыми учебными программными средствами.

Технология решения задач с использованием компьютера

Изучение материала дает возможность учащимся:

- понимать назначение и содержание основных звеньев технологической цепочки: «объект — постановка задачи — построение информационной модели — построение алгоритма — компьютерная реализация — получение и анализ результата»;
- уметь правильно подобрать программно-аппаратные средства, необходимые для решения поставленной задачи;
- научиться реализовывать учебно-исследовательские проекты.

Уровень обязательной подготовки определяется следующим требованием:

- знать структуру основной технологической цепочки, содержание и назначение ее звеньев.

Социально-мировоззренческий аспект

Уровень обязательной подготовки определяется следующими требованиями:

- знать содержание процесса информатизации общества и роль в нем средств вычислительной техники;
- понимать смысл и значение информационной точки зрения на мир.

Литература

1. Белошапка В. К. О языках, моделях и информатике // Информатика и образование. 1987. № 6.
2. Белошапка В. К. Три аспекта мироздания или мир как информационная структура // Информатика и образование. 1988. № 5.
3. Белошапка В. К., Лесневский А.С. Основы информационного моделирования // Информатика и образование. 1989. № 3.
4. Белошапка В. К., Лесневский А.С. Новые тенденции и новые опасности в образовании в век информатики: Доклад на конференции «Дети в век информатики», Болгария, Албена, 1991.
5. Белошапка В. К., Лесневский А.С. Сивол — инструментальная система для моделирования: Тезисы доклада на советско-английском коллоквиуме по моделированию, Москва, 1990.
6. Белошапка В. К. Информатика как наука о буквах // Информатика и образование, 1991 № 6.
7. Белошапка В. К., Лесневский А.С. Информационное моделирование — ведущий метод современного естествознания: Доклад на конференции «Разработка и использование педагогических программных средств», Москва, 1991.
8. Белошапка В. К. Информационное моделирование в примерах и задачах. Омск: Изд. Омского пед. ин-та, 1992. С. 1-160.

Продаю программы для компьютеров «Агат-7», «Агат-9», «APPLE-][», «Правец-8С», «Микроша», «Апогей», «Радио-86РК» (32кБ), MSX-1,2 («Ямаха»), COMMODORE-500.

Каталог высылается бесплатно. В заявку вложите конверт с Вашим адресом и укажите тип ПЭВМ.

Адрес: 140106, Московская область, г. Раменское-6, а/я 60.
 Контактный телефон: 3-84-88
 Код из г. Москвы: 8-(246)
 Код из других городов: 8-(09646)

М. Е. Степанов

ЕДИНЫЙ МЕТОДИЧЕСКИЙ ПОДХОД К ПОСТРОЕНИЮ ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ

Линейные отображения и наклонный эллипс

Статья продолжает публикацию «Единый методический подход к построению сложных компьютерных изображений» (ИНФО. 1993. №1). Предыдущая публикация ориентирована на язык Бейсик-Агат. Теперь наши исследования относятся к MSX-Бейсик для ПЭВМ «Ямаха», однако описываемые методы легко реализуются на любом языке с графическим расширением.

1. Равномерное движение и линейные отображения

Начнем с вывода основной формулы, используя при этом наглядную кинематическую модель линейного отображения квадрата на параллелограмм.

Через квадратное окно на пол темной комнаты падает световой пучок, который освещает на полу область в форме параллелограмма (рис. 1). По стеклу ползет муха, а ее тень перемещается по параллелограмму. Если муха движется параллельно краю окна, то ее тень перемещается параллельно одной из сторон параллелограмма.

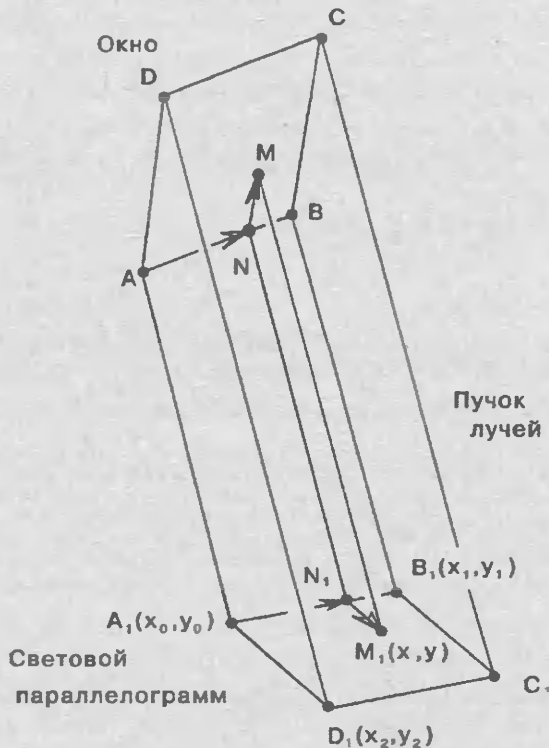


Рис. 1

Будем считать, что сторона квадрата имеет единичную длину, а положение любой точки внутри квадрата задается парой координат (t_1, t_2) . Положение параллелограмма на плоскости задается координатами трех вершин. Если муха переместилась по отрезку AB от A до N , то, как мы убедились ранее, координаты ее тени N_1 выражаются по формулам

$$X_N = X_0 + t_1 * (X_1 - X_0)$$

$$Y_N = Y_0 + t_2 * (Y_1 - Y_0)$$

При движении от N до M тень движется от N1 к M1 параллельно стороне, вершины которой заданы координатами (X0, Y0) и (X2, Y2). Применяя ту же формулу, получаем, что

$$X = X_N + t_1 * (X_2 - X_0)$$

$$Y = Y_N + t_1 * (Y_2 - Y_1)$$

Исключив промежуточные координаты, получаем формулы для линейного отображения квадрата на параллелограмм

$$X = X_0 + t_1 * (X_1 - X_0) + t_2 * (X_2 - X_0)$$

(1)

$$Y = Y_0 + t_1 * (Y_1 - Y_0) + t_2 * (Y_2 - Y_0)$$

Задание 1.1. Построить на экране решетку из точек, равномерно распределенных в параллелограмме (рис. 2).

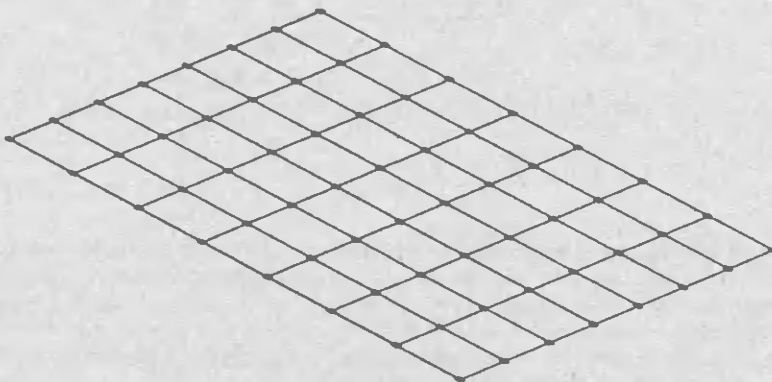


Рис. 2

Решение. Основной принцип использования формул (1) состоит в том, что нужная структура строится в квадрате, а затем переносится в параллелограмм. В квадрате равномерная решетка строится с использованием двух вложенных циклов с параметрами t_1 и t_2 . В подпрограмме эти величины преобразуются в экранные координаты X, Y.

10 X0=127:Y0=95:X1=191:

Y1=191:X2=191:Y2=48

20 SCREEN 2:

'ВКЛЮЧЕНИЕ ГРАФИЧЕСКОГО РЕЖИМА

30 FOR T1=0 TO 1 STEP 1/16

40 FOR T2=0 TO 1 STEP 1/8

50 GOSUB 1000:' ВЫЧИСЛЕНИЕ X И Y

60 NEXT T2,T1

70 GOTO 70:' ЗАДЕРЖКА РИСУНКА НА ЭКРАНЕ

1000 X=X0+T1*(X1-X0)+T2*(X2-X0)

1010 Y=Y0+T1*(Y1-Y0)+T2*(Y2-Y0)

1020 PSET(X,Y):' УСТАНОВКА ТОЧКИ

1030 RETURN

Отметим, что формулы (1) на самом деле отображают не только квадрат на параллелограмм, но и всю координатную плоскость (t_1, t_2) на плоскость (X, Y). При этом сеть квадратов переходит в сеть параллелограммов (чтобы представить это более наглядно, нужно заменить окно и непрозрачную стену на сложенный из квадратных блоков витраж). Таким образом, в

формулах (1) можно использовать любые значения t_1 и t_2 . Для выбора границ изменения этих переменных нужно мысленно покрыть плоскость X, Y сетью параллелограммов и ориентироваться по ней.

Задание 1.2. Используя программу из предыдущего задания, покройте весь экран точечной решеткой.

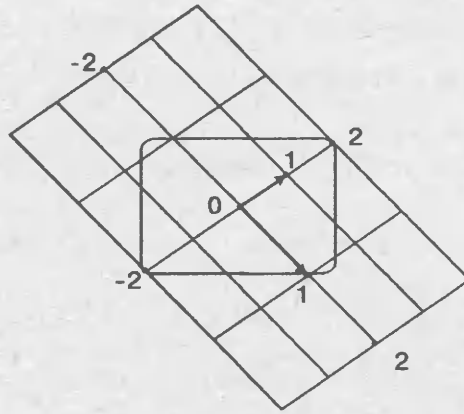


Рис. 3

Решение. На рис.3 изображено покрытие экрана параллелограммами, которое позволяет распространить решетку на весь экран. При этом достаточно изменить пределы, в которых меняются параметры t_1 и t_2 на числа -2 и 2 (строки 30 и 40 программы). В строке 1020 можно использовать условный оператор, чтобы исключить установку точки вне экрана. Поскольку значительное количество точек ложится вне экрана, то при работе программы появление первой точки на экране произойдет с заметным запозданием.

Задание 1.3. Построить «косую» шахматную доску.

Решение. Построение клеток полностью аналогично программе 1.3*. Для закраски «черных» (на экране они могут оказаться белыми) следует указать исходную точку для оператора закраски. Именно тут и используются формулы (1). Если среди графических средств языка нет оператора закраски, нужно «заштриховать» или «зататуировать» соответствующую клетку точками, что усложняет программу, но может быть проделано с помощью тех же формул.

```

10 DATA 50,120,80,10,170,180
20 READ X0,Y0
30 READ X1,Y1
40 READ X2,Y2
50 K1=X1-X0:M1=X2-X0
60 K2=Y1-Y0:M2=Y2-Y0
70 SCREEN 2
80 FOR T2=0 TO 1 STEP 1/10
90 FOR T1=0 TO 1
100 X=X0+T1*K1+T2*M1
110 Y=Y0+T1*K2+T2*M2
120 IF T1=0 THEN PSET(X,Y) ELSE
      LINE-(X,Y):' ПРОВЕДЕНИЕ ОТРЕЗКА
130 NEXT T1,T2
140 FOR T1=0 TO 1 STEP 1/10

```

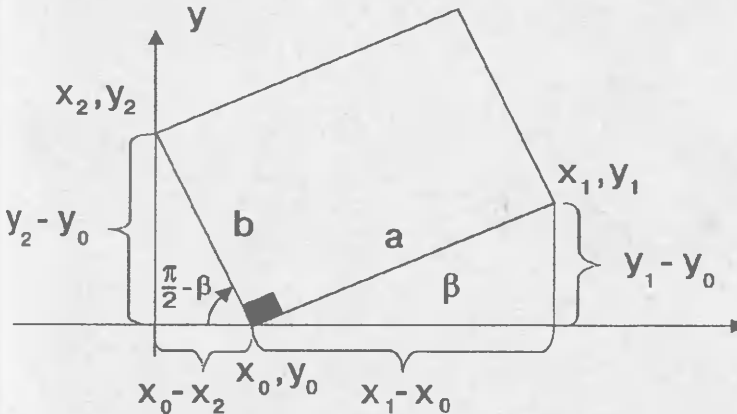
* ИНФО. 1993. № 1.


```

150 FOR T2=0 TO 1
160 X=X0+T1*K1+T2*M1
170 Y=Y0+T1*K2+T2*M2
180 IF T2=0 THEN PSET(X,Y)
      ELSE LINE-(X,Y)
190 NEXT T2,T1
200 P=1
210 FOR T1=0 TO 19/20 STEP 1/10
220 FOR T2=0 TO 19/20 STEP 1/10
230 X=X0+(T1+1/20)*K1+(T2+1/20)*M1
240 Y=Y0+(T1+1/20)*K2+(T2+1/20)*M2
250 IF P=-1 THEN PAINT(X,Y): ' ЗАКРАСКА
260 P=-P
270 NEXT T2
280 P=-P
290 NEXT T1
300 GOTO 300

```

Задание 1.4. Построить точечную решетку, размещенную в прямоугольнике, наклоненном под углом β к горизонтали (рис. 4).



$$y_2 - y_0 = b \cdot \cos \beta$$

$$x_0 - x_2 = b \cdot \sin \beta$$

$$x_1 - x_0 = a \cdot \cos \beta$$

$$y_1 - y_0 = a \cdot \sin \beta$$

Рис. 4

Решение. Отличие от задания 1.1 состоит только в том, что мы можем произвольно выбрать координаты X_0 , Y_0 , угол β и длины сторон прямоугольника, а координаты X_1 , Y_1 и X_2 , Y_2 должны выражаться через эти величины:

$$X_1 = X_0 + A \cdot \cos \beta$$

$$Y_1 = Y_0 + A \cdot \sin \beta$$

$$X_2 = X_0 - B \cdot \sin \beta$$

$$Y_2 = Y_0 + B \cdot \cos \beta$$

(2)

Перейдем от точечных решеток к построению линий. Если начертить на стекле темную линию, то в светлом параллелограмме на полу появится ее тень, — преобразованная линия. Легко понять, что прямая линия преобразуется в прямую.

Задание 1.5. На плоскости (t_1 , t_2) нарисована ломаная линия. Отобразите ее в параллелограмм.

Решение. Зададим линию с помощью целочисленных координат ее вершин (рис. 5). В программе приведем эти координаты к долям единичного отрезка, т. е. мысленно переместим рисунок в единичный квадрат. Затем последовательно будем отображать вершины в параллеле-



Рис. 5

лограмм, соединяя полученные точки отрезками. Более того, мы отобразим ломаную в два параллелограмма, чтобы показать применение формул (1) для построения симметричных изображений.

```

10 DATA 8,0,7,4,7,8,2,20,2,
      22,3,23,4,22,9,14
20 DATA 10,14,9,20,9,28,10,
      29,11,29,12,27,12,21,13,16
30 DATA 14,16,14,21,13,30,
      14,31,15,31,16,30,17,24,18,16
40 DATA 19,16,21,28,22,29,
      23,29,24,28,23,16,23,9,24,8
50 DATA 26,10,28,14,30,15,
      31,14,30,10,28,6,22,0

60 N=39
70 DIM T1(N),T2(N)
80 FOR I=1 TO N
90 READ R1,R2
100 T1(I)=R1/N:T2(I)=R2/N
110 NEXT I
120 SCREEN 2
130 X0=63:Y0=191:X1=127
      Y1=191:X2=0:Y2=0

140 FOR J=1 TO 2
150 FOR I=1 TO N
160 X=X0+T1(I)*(X1-X0)+T2(I)*(X2-X0)
170 Y=Y0+T1(I)*(Y1-Y0)+T2(I)*(Y2-Y0)
180 IF I=1 THEN PSET(X,Y)
      ELSE LINE-(X,Y)

190 NEXT I
200 X0=191:X2=255
210 NEXT J
220 GOTO 220

```

Задание 1.6. Впишите в параллелограмм овал.

Решение. Используем параметрическое представление окружности на плоскости (t_1, t_2). Изменяя параметр (угол), вычисляем значения t_1 и t_2 (рис. 6), а затем преобразуем их по формулам (1).

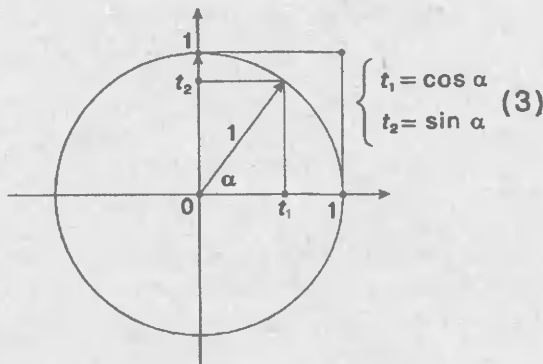


Рис. 6

```

10 SCREEN 2:PI=4*ATN(1)
20 X0=128:Y0=95:X1=223:
      Y1=71:X2=159:Y2=23
30 FOR I=0 TO 36
40 T1=COS(PI*I/18)
50 T2=SIN(PI*I/18)
60 X=X0+T1*(X1-X0)+T2*(X2-X0)
70 Y=Y0+T1*(Y1-Y0)+T2*(Y2-Y0)
80 IF I=0 THEN PSET(X,Y)
      ELSE LINE-(X,Y)
90 NEXT I
100 GOTO 100

```

На экране будет начерчен овал. Можно доказать, что на самом деле он является эллипсом.

Задание 1.7. Разработайте способ построения наклонного эллипса по углу наклона и длине полуосей A и B .

Решение. Чтобы решить задачу, следует отобразить окружность в наклонный прямоугольник, точно такой же, как в задании 4.4. Для этого достаточно подставить в (1) значения из (2) и (3), используя рис. 6. В итоге получаем формулы:

$$X = X_0 + A \cdot \cos\alpha \cdot \cos\beta - \sin\alpha \cdot \sin\beta \quad (4),$$

$$Y = Y_0 + A \cdot \cos\alpha \cdot \sin\beta + B \cdot \sin\alpha \cdot \cos\beta,$$

которые позволяют строить наклонный эллипс, меняя в цикле параметр α .

Из предыдущей статьи ясно, что формулы (1) и (4) открывают путь для модификации соответствующих изображений. Введение формулы (1) в подпрограмму обработки координат позволяет отобразить изображения в параллелограмм, «перекосить» их, повернуть, отразить симметрично. Использование формул (4) при построении объемных изображений позволяет «наклонить» их. Продемонстрируем это на примерах (без использования подпрограмм).

Задание 1.8. Построить однополостный гиперболоид в наклонном положении (сравни с заданием 2.8)*.

Решение.

```

10 PI=4*ATN(1):CLS:

```

* ИНФО. 1993. № 1.

```

PRINT »СЧЕТ»:DIM XT(18),YT(18)
20 A=1:B=.5:AL=-PI/4:CA=COS(AL):
    SA=SIN(AL):'В ПРОГРАММЕ AL - УГОЛ
    НАКЛОНА ЭЛЛИПСА; U - УГОЛ-ПАРАМЕТР
30 FOR I=1 TO 18:U=1E-03+I*PI/9:
    CU=COS(U):SU=SIN(U)
40 XT(I)=A*CA*CU-B*SA*SU:
    YT(I)=A*SA*CU+B*CA*SU
50 NEXT I
60 SCREEN 2:C1=50:C2=50:RV=40:
    D1=150:D2=150:RN=55
70 FOR I=1 TO 18
80 PSET(RV*XT(I)+C1,RV*YT(I)+C2)
90 J=I+6:IF J18 THEN J=J-18
100 LINE-(RN*XT(J)+D1,RN*YT(J)+D2):
    NEXT I
110 GOTO 110

```

Задание 1.9. Отобразите сердечко (задание 1.9) на грани куба (рис. 7).

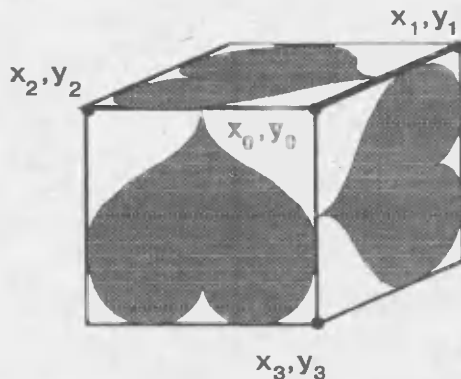


Рис. 7

Решение. В программе использована единая параметризация сердечка: параметр, пробегая значения от 0 до 1, очерчивает всю кривую. То же самое можно было бы проделать и с помощью кусочной параметризации.

```

10 DATA 128,96,218,6,38,96,128,186
20 DIM X(3),Y(3),K1(3),M1(3),K2(3),
    M2(3),XE(3),YE(3),XS(2,3),YS(2,3)
30 FOR I=0 TO 3
40 READ X(I),Y(I)
50 NEXT I
60 FOR I=1 TO 3
70 K1(I)=X(I)-X(0):K2(I)=Y(I)-Y(0)
80 J=I+1:IF J3 THEN J=1
90 M1(I)=X(J)-X(0):M2(I)=Y(J)-Y(0)
100 NEXT I
110 XE(0)=.25:YE(0)=0
120 XE(1)=.25:YE(1)=2/3
130 XE(2)=.75:YE(2)=2/3

```

```

140 XE(3)=.75:YE(3)=0
150 SCREEN 2:PI=4*ATN(1)
160 FOR I=1 TO 3
170 LINE(X(0),Y(0))-(X(I),Y(I))
180 XX=X(0)+K1(I)+M1(I)
190 YY=Y(0)+K2(I)+M2(I)
200 LINE-(XX,YY)
210 J=I+1:IF J3 THEN J=1
220 LINE-(X(J),Y(J))
230 LINE-(X(0),Y(0))
240 NEXT I
250 FOR T=0 TO 1 STEP 1/80
260 IF T/8 THEN J=0:U=4*T*PI
270 IF T=1/8 AND T/2 THEN
      J=1:U=4*(.375-T)*PI+PI/2
280 IF T=1/2 AND T/8 THEN
      J=2:U=4*(.75-T)*PI
290 IF T=7/8 THEN
      J=3:U=4*(T-.875)*PI+PI/2
300 FOR I=1 TO 3
310 T1=XE(J)+(1/4)*COS(U)
320 T2=YE(J)+(1/3)*SIN(U)
330 XS(1,I)=X(0)+K1(I)*T1+M1(I)*T2
340 YS(1,I)=Y(0)+K2(I)*T1+M2(I)*T2
350 IF T0 THEN LINE (XS(2,I),YS(2,I))
      -(XS(1,I),YS(1,I))
360 XS(2,I)=XS(1,I):YS(2,I)=YS(1,I)
370 NEXT I,T
380 FOR I=1 TO 3
390 XX=X(0)+K1(I)*XE(1)+M1(I)*YE(1)
400 YY=Y(0)+K2(I)*XE(1)+M2(I)*YE(1)
410 PAINT(XX,YY)
420 NEXT I
430 GOTO 430

```

Задание 1.10. Изобразите сердечко, обведенное широкой полосой (рис. 8).

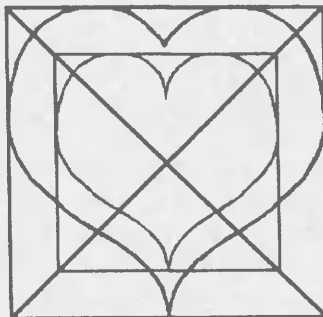


Рис. 8

Решение. Поочередно отобразим сердечко в большой и малый параллелограммы.

Построение поверхностей

2. Поверхности и формулы

Задача построения поверхности, заданной формулой, которая выражает функцию через две независимые переменные, сходна с задачей построения графика обычной функции одной переменной.

Задание 2.1. Используя формулы линейного отображения, отобразите график функции в параллелограмм на экране.

Решение. Целостную картину графика функции можно себе представить, если вообразить всю плоскость. Однако на экран можно перенести только часть плоскости. Вырежем из воображаемой бесконечной плоскости прямоугольник со сторонами, параллельными осям координат, так, чтобы внутри этого прямоугольника график обладал всеми характерными особенностями. Сожмём этот прямоугольник до размеров единичного квадрата, а затем отобразим квадрат на экранном параллелограмме (рис. 9).

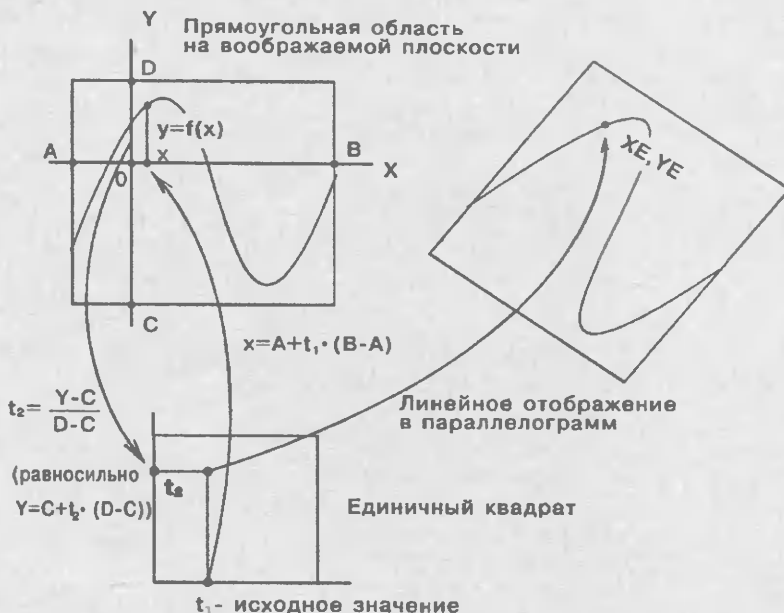


Рис. 9

```

10 SCREEN 2
20 DEF FNF(X)=SIN(X)
30 A=-8:B=8:C=-1:D=1
40 N=100
50 X0=0:Y0=90:X1=100:Y1=180:X2=150:Y2=0
60 FOR T1=0 TO 1 STEP 1/N
70 X=A+T1*(B-A):Y=FNF(X):T2=(Y-C)/(D-C)
80 XE=X0+T1*(X1-X0)+T2*(X2-X0)
90 YE=Y0+T1*(Y1-Y0)+T2*(Y2-Y0)
100 IF T1=0 THEN PSET(XE,YE)
      ELSE LINE-(XE,YE)
110 NEXT T1
120 GOTO 120
  
```


Для объяснения идей, лежащих в основе построения поверхностей, снова используем модель. Представим себе щетку с прямоугольным основанием и плотной щетиной, перпендикулярной плоскому деревянному основанию. В целом щетка выглядит как параллелепипед и может быть изображена на плоскости следующим образом: основание можно изобразить с помощью параллелограмма, основания ворсинок как равномерную решетку в параллелограмме, а сами ворсинки как вертикальные отрезки одинаковой длины (рис. 10).

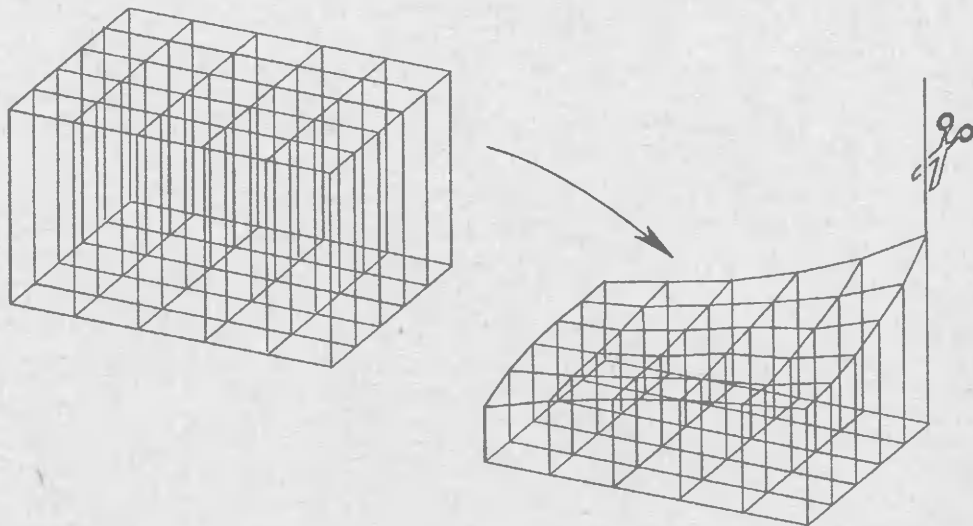


Рис. 10

Если сделать щетке эффектную стрижку, концы ворсинок образуют поверхность. Математическим средством, которое заменяет ножницы, является формула, задающая функцию от двух переменных. Положение ворсинки на основании щетки задается двумя величинами, а длина ворсинки выражается через эти величины.

Задание 2.2. Построить изображение поверхности, заданной с помощью формулы $Z = 1/(X^2 + Y^2 + 1)$.

Решение. Поверхность будем строить в виде сетки с четырехугольными ячейками. Сначала вычислим координаты концов ворсинок и сохраним их в массивах, а затем построим сетку. На вычисление координат уйдет определенное время, так что построение начнется с задержкой. В деталях программы поможет разобраться рис.11.

```
10 N=20:H=120:A=4:B=-4
```

```
20 X0=0:Y0=150:X1=130:
```

```
Y1=190:X2=110:Y2=110
```

```
30 DIM X(N,N),Y(N,N)
```

```
40 DEF FNF(X,Y)=1/(X^2+Y^2+1)
```

```
50 FOR I=0 TO N
```

```
60 FOR J=0 TO N
```

```
70 T1=I/N:T2=J/N
```

```
80 XT=-A+T1*2*A
```

```
90 YT=B-T2*B
```

```
100 Z=FNF(XT,YT)
```

```
110 X(I,J)=X0+T1*(X1-X0)+T2*(X2-X0)
```

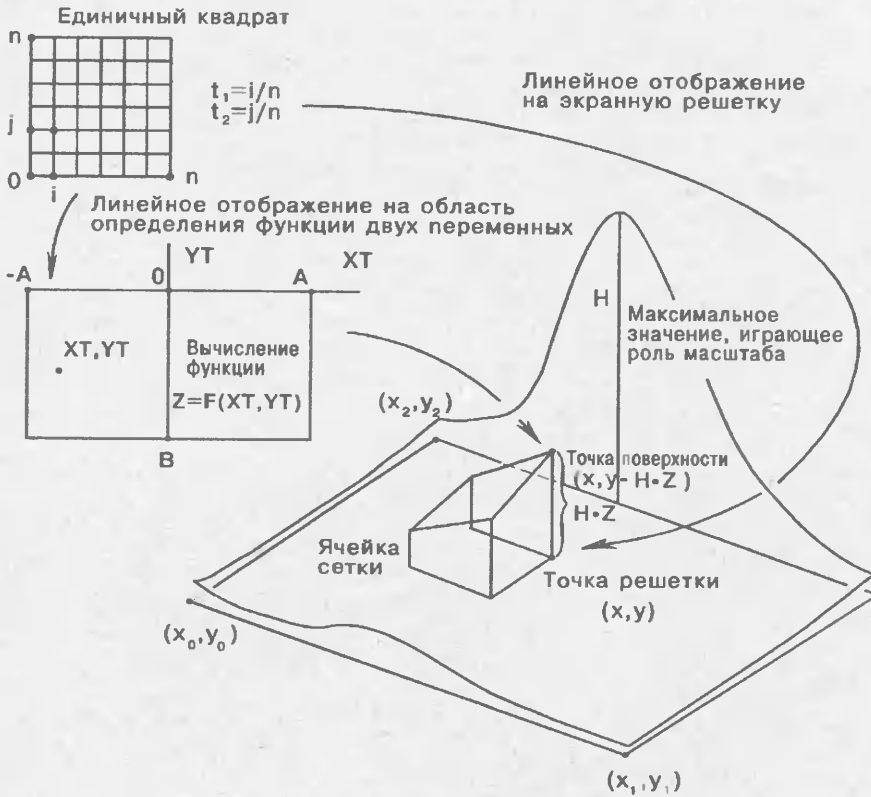


Рис. 11

```

120 Y(I,J)=Y0+T1*(Y1-Y0)+T2*(Y2-Y0)-H*Z
130 NEXT J,I
140 SCREEN 2
150 FOR I=0 TO N-1
160 FOR J=0 TO N-1
170 PSET(X(I,J),Y(I,J))
180 LINE-(X(I+1,J),Y(I+1,J))
190 LINE-(X(I+1,J+1),Y(I+1,J+1))
200 LINE-(X(I,J+1),Y(I,J+1))
210 LINE-(X(I,J),Y(I,J))
220 NEXT J,I
230 GOTO 230

```

3. Конструктивное построение поверхностей

Линейные отображения помогают строить поверхности, которые трудно задать формулами, но достаточно просто представить конструктивный способ их построения.

Задание 3.1. Построить изображение листа Мебиуса.

Решение. Представим себе, что прямоугольник вращается вокруг одной из своих сторон как вокруг оси (рис. 12). На плоскости можно изобразить некоторые фазы его вращения. Ясно, что две его вершины описывают эллипсы. Этот факт позволяет построить изображение фазы движения по углу поворота прямоугольника.

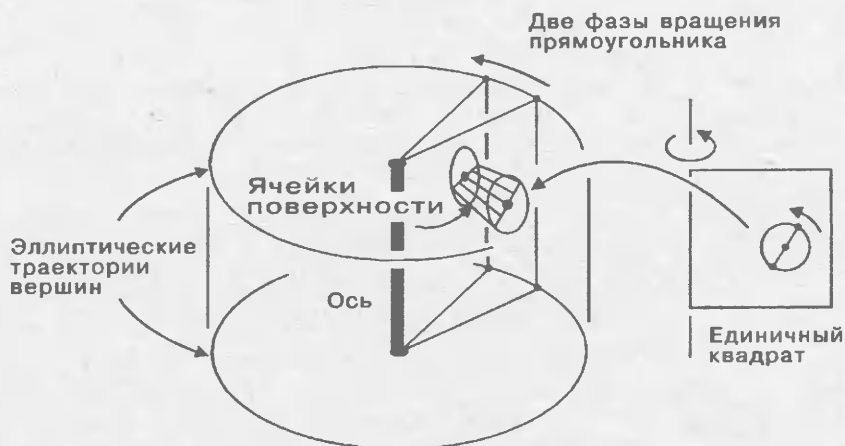


Рис. 12

В плоскости прямоугольника вокруг закрепленной середины вращается отрезок. Его концы описывают окружность. Представим себе, что такой же отрезок вращается внутри неподвижного единичного квадрата. Задав угол его наклона, можно вычислить координаты его концов в единичном квадрате, а затем линейно отобразить в параллелограмм, изображающий собой фазу движения прямоугольника. Таким образом можно построить изображение отрезка в каждой фазе движения прямоугольника.

Отрезок при движении заметает некоторую поверхность. Положим, что угол поворота отрезка всегда равен половине угла поворота прямоугольника. Тогда после полного оборота прямоугольника отрезок повернется на половину оборота и совместится сам с собой, но перевернувшись. В этом случае заметаемая поверхность и будет листом Мебиуса. Будем изображать лист Мебиуса в виде сетки. Для этого разобьем отрезок на равные части и вычертим не только фазы движения отрезка, но и траектории точек, которые делят отрезок на части.

```

10 SCREEN 2:PI=4*ATN(1):X0=125:
    Y0=150:Y1=50:R=80:R0=.25
20 K=.5:N=36:M=6:DIM X(1,M),Y(1,M)
30 FOR I=0 TO N:U1=I*PI/18:U2=U1/2
40 XT=X0+R*COS(U1):YT=Y0-K*R*SIN(U1)
50 T1=.75+.25*COS(U2):
    T2=.75-.25*SIN(U2):
    T3=.75-.25*COS(U2):
    T4=.75+.25*SIN(U2)
60 XA=X0+T1*(XT-X0):
    YA=Y0+T1*(YT-Y0)+T2*(Y1-Y0)
70 XB=X0+T3*(XT-X0):
    YB=Y0+T3*(YT-Y0)+T4*(Y1-Y0)
80 LINE(XA,YA)-(XB,YB):FOR J=0 TO M
90 X(0,J)=XA+(J/M)*(XB-XA):
    Y(0,J)=YA+(J/M)*(YB-YA):NEXT J
100 IF I=0 THEN 120
110 FOR J=0 TO M:
    LINE(X(0,J),Y(0,J))-(X(1,J),Y(1,J)):
    NEXT J
120 FOR J=0 TO M:

```

```

X(1,J)=X(0,J):Y(1,J)=Y(0,J)
130 NEXT J,I
140 GOTO 140

```

Задание 3.2. Построить изображение поверхности вращения. В качестве образующей возьмите график функции.

Решение. Задача решается аналогично предыдущей. Вместо вращающегося отрезка используется неподвижная (относительно прямоугольника) кривая. Ее построение для каждой фазы движения проводится так же, как в задаче 2.1.

```

10 SCREEN 2:PI=4*ATN(1)
20 N=36:M=20
30 DIM T1(M),T2(M),X(1,M),Y(1,M)
40 X0=125:Y0=150:Y1=50:R=100:K=.5
50 FOR J=0 TO M
60 T1(J)=.1+(J/M)*.9
70 T2(J)=(10-1/T1(J))/9
80 NEXT J
90 FOR I=0 TO N
100 U=I*PI/18
110 XT=X0+R*COS(U):YT=Y0-K*R*SIN(U)
120 FOR J=0 TO M
130 X(0,J)=X0+T1(J)*(XT-X0):
Y(0,J)=Y0+T1(J)*(YT-Y0)+T2(J)*(Y1-Y0)
140 IF J=0 THEN PSET(X(0,J),Y(0,J))
ELSE LINE-(X(0,J),Y(0,J))
150 NEXT J
160 IF I=0 THEN 190
170 FOR J=0 TO M
180 LINE(X(0,J),Y(0,J))-(X(1,J),Y(1,J))
NEXT J
190 FOR J=0 TO M:
X(1,J)=X(0,J):Y(1,J)=Y(0,J)
200 NEXT J,I
210 GOTO 210

```

Более подробно с предлагаемой методикой построения изображений можно ознакомиться по работе автора «Единый методический подход к построению сложных компьютерных изображений (с программами на turbo-BASIC для IBM PC)».

Предлагаем программные средства и периферию для БК-0010(01), БК-0011М, УКНЦ (МС-0511), Синклер-128, Спектрум, IBM-совместимых компьютеров.

Для получения каталога вышлите конверт с марками по адресу:

189510, Санкт-Петербург, Ломоносов, а/я 649. «КИ-ЧПМ».

Н. Ю. Пахомова

ДЕЛОВЫЕ ИГРЫ НА УРОКЕ ИНФОРМАТИКИ

Компьютерные игры интересны, некоторые из них полезны, развивают, обучают и тренируют. Но так называемая деловая игра, где нужно моделировать обстановку делового партнерства, конкуренции и сотрудничества, оказывается более привлекательной даже при гипотетическом (воображаемом) использовании компьютера.

Игра «Коммивояжер»

Дети всегда готовы играть и с нетерпением ждут объявления правил игры. Предлагаю им образовать фирмы, разбившись на группы по 3—4 человека. Все фирмы получают одно и то же задание на программирование. Даже если учащиеся научились только составлять алгоритмы и записывать их на алгоритмическом языке, играть уже можно.

Начиная игру каждая фирма сама выбирает того, кто у нее будет Президентом, кто Программистом-аналитиком (главным мыслителем), кто Дизайнером-оформителем. Четвертое лицо, Коммивояжера, я прошу разрешить выбрать мне, так как стараюсь назначить самого слабого по предмету.

Решив задачу, т. е. написав программу, фирма оформляет свой программный продукт на обычном тетрадном листе: записывает программу, снабжает ее инструкцией, рекламой, рисует фирменные знаки и т. п. Затем каждая фирма должна подготовить своего коммивояжера для поездки в другую фирму с целью продажи программы. Задача фирмы объяснить ему суть программы, подготовить его, чтобы никакие неожиданные, каверзные вопросы конкурирующей фирмы не застали врасплох. От хорошей работы коммивояжера на заключительном этапе зависит успех фирмы.

И вот игра началась. Совсем не праздное дело — распределить роли. Тихая деловая пауза переходит в следующую волну обсуждений и споров о названии фирмы, ее эмблеме, потом опять рабочая тишина.

Группы располагаются так, чтобы не мешать друг другу и до поры не разглашать своих «секретных» замыслов. Разделение ролей весьма условно — все в той или иной мере участвуют в программировании, все активно обсуждают, вносят предложения.

Когда все фирмы будут готовы для обмена коммивояжерами, принимающая фирма тщательно проверяет программу, задает вопросы по существу: почему сделано так, а не иначе, просит разъяснить некоторые места программы. Здесь коммивояжеру придется выдерживать экзамен и постоять за честь своей фирмы.

На заключительном этапе фирмы играют в куплю-продажу. Покупающая фирма составляет рецензию, предлагая форму расчета и денежные единицы. Нам едва хватает двоедного урока. Ребята расходятся, предлагая для следующей игры свои варианты правил.

На память у учителя остаются тетрадные листочки с текстами программ, зарисованные символически и красочными названиями фирм. Вот фирма «Москва-сортiroвочная» представляет программу сортировки методом «пузырька». Ее конкуренты из фирмы «Одуванчик», или «Dandelion» (это совместное советско-американское предприятие), в рекламе своей программы сортировки предлагают использовать ее и в банковских операциях, и при построении солдат по росту. Пожалуй, превзошла всех фирма «Abak computer company», предлагая — «Новый программный продукт сортировки для зоопарков». Они пишут: «Продукция фирмы «Абак» хорошо зарекомендовала себя на мировом рынке. И теперь лучшие программисты нашей фирмы предлагают вам высокоэффективную, быстродействующую программу сортировки диких обезьян!...»

Рецензии отражают характеры их авторов, полны юмора и фантазии. Фирма «Москва-сортiroвочная» получила суровый приговор: «Ваша программа содержит три ошибки. Кроме того, она узкоспециализирована». В другой рецензии ощущается экономическое положение страны — процветает бартер и натуральный обмен: «Цена программы: три машины картошки, две машины свеклы, 184 батона белого хлеба, 241 буханка черного хлеба, две красивые девушки и пять красивых юношей».

В подобную игру играют со своими учениками и мои коллеги из других школ. Игры очень похожи, но есть и отличия. Иногда учитель может одновременно быть заказчиком программы и ее покупателем. Условные деньги накапливаются в условном банке, фирмы вольны либо потратить их на отметку в журнал для какого-либо члена фирмы (они сами решают, для кого), либо купить у конкурирующей фирмы необходимый ей программный модуль. Компьютер используется для отладки написанных программ.

Предела варьирования и совершенствования правил игры нет. Важно то, что на уроках информатики такие деловые игры, преследующие обучающие цели, результат учебного труда воплощают в социально значимый продукт, проигрываются профессиональные амплуа, где компьютеру отводится место помощника.

У каждого учителя вместе с опытом работы накапливаются в его арсенале подобные игровые организационные формы обучения, позволяющие эмоционально раскрасить урок. Не секрет, что игровые формы там, где это возможно, предпочтительнее. Поэтому, когда творчество учителя приводит к находке какой-либо новой игровой формы — этот опыт заслуживает внимания учителя. Хоча предложить несколько своих находок.

Игра «Найди ошибку»

Работу на закрепление знаний, например, можно организовать так. Текст программы, написанный учителем на доске, ребята переписывают на отдельный тетрадный листок. Учитель просит их сделать умышленно в этом тексте ошибки и хорошо их завуалировать, не подписывая работу. Необходимо только подсчитать и записать на том же листочке общее количество сделанных в ней ошибок. Затем программа с доски стирается. А все листочки с ошибочными программами-заготовками собираются в один пакет. После тщательного перемешивания (как подобает в настоящей лотерее) каждый ученик вытаскивает один листок. Работа по распознаванию ошибок увлекает всех. Бывает весьма курьезная ситуация, когда найденных ошибок оказывается больше, чем было объявлено автором программы-заготовки.

Игра «Робот в лабиринте»

Эту игру можно использовать, делая первые шаги в алгоритмизации. В ее основе лежит задача, известная многим: необходимо провести по любому лабиринту без кольцевых коридоров робота, который умеет выполнять только несколько действий: шаг вперед; поворот вправо на 90° , поворот влево на 90° , узнать, есть ли стена впереди на расстоянии шага; вышел ли робот из лабиринта.

Для того чтобы представить какой-либо из простейших лабиринтов этой задачи, можно воспользоваться картинкой, приведенной в книге Ж.-П. Ла-

муатье «Упражнения по программированию на Фортране IV» (рис. 1).

Решением данной задачи должен стать алгоритм прохождения робота по лабиринту. Дети еще не знают, что такое алгоритм, блок-схема алгоритма. Не знают и хитростей обхода лабиринта: нужно идти, держась все время за одну стену (или правую, или левую).

Для записи придуманного алгоритма учитель показывает ученикам четыре основных элемента для построения блок-схем (рис. 2).

Итак, для игры у нас есть: задача, средства записи ее решения и, конечно, желание играть. Нет пока еще правил игры. Но они станут понятны по ходу рассказа.

Сначала учитель предлагает детям для решения задачу, объясняя, как можно записать инструкцию для робота, как для проверки исполнить ее воображаемым роботом.

Когда готово решение, учитель предлагает одному из учеников нарисовать блок-схему алгоритма на доске и показать всем. В классе сразу находится несколько оппонентов из числа претендентов на право первым показать свой алгоритм. Когда на доске готовы и лабиринт, и блок-схема алгоритма, начинается действие игры. Дети внимательно следят за тем, как автор будет отдавать роботу команды, показывая указкой, какую из них он читает

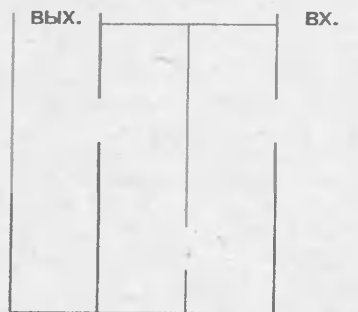


Рис. 1

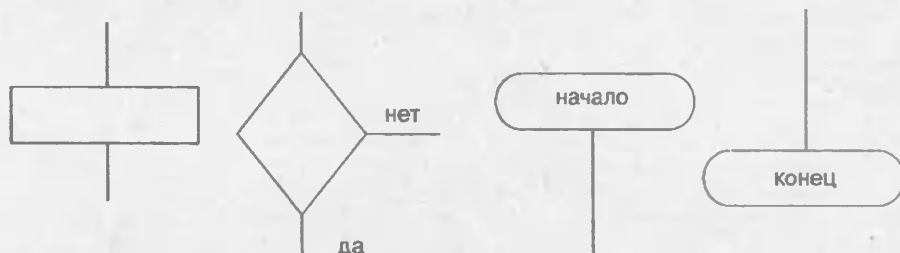


Рис. 2

на блок-схеме. Оппонента мы вооружаем чем-нибудь, напоминающим робота, и просим, поставив робота в начало лабиринта, четко выполнять те команды, которые будут поступать от автора алгоритма.

Ребята, как нейтральные арбитры, тут же исправляют ошибки или сбой оппонента, если он повернул робота не в ту сторону, перешел к команде не по стрелке следования в блок-схеме. Идет «защита проекта».

Заканчивается она либо поражением автора, когда, к радости всех «противников», по его команде робот разбивается о стену или «топчется на месте», выполняя повторяющиеся действия, либо алгоритм приводит к «победе» — и робот благополучно выходит из лабиринта. Начинаем следующий раунд «защиты» проекта нового автора и оппонента.

В ходе игры дети хорошо усваивают основные моменты, связанные с понятием алгоритма. Они различают, что такое исполнитель алгоритма, действия исполнителя. Они начинают понимать даже основные алгоритмические конструкции: альтернативы, повторения, линейные последовательности. Учитель при удобной ситуации, возникающей во время игры, старается заострить их внимание на тех или иных понятиях, объектах. Заметим, что новые знания, полученные в игре, грамотно акцентированные и обобщенные, дети усваивают лучше, глубже, уже готовыми к актуализации.

Игра «Автомат “Ленивые вареники”»

Игра в разработку и защиту проекта автомата по производству ленивых вареников очень похожа на предыдущую. Ленивые вареники — это тесто с творогом в однородной массе, раскатанное колбаской и разрезанное затем на небольшие доли, напоминающие пельмени, сваренное в подсоленной воде и поданное со сметаной. Для этой кулинарной технологии (она проста для понимания операций и их последовательности) дети легко придумывают воображаемые устройства.

Это не важно, что проекты этих устройств далеки от уровня, необходимого для выполнения их в металле, нет стройных чертежей и необходимых расчетов. Важно, что дети продумывают возможность автоматизации вполне осуществимого процесса, составляют алгоритмы технологии реального производства. Такая задача позволяет расширить представление о природе понятия алгоритма, об исполнителе алгоритма и его возможностях.

Эта игра увлекает своей областью творчества — конструированием. Включаются пространственное воображение, весь житейский опыт, знания и представления из физики, математики, черчения. Установку проектируем для школьной столовой — это интересно всем. Производим подсчет производительности такого автомата и отвечаем на вопрос: как организовать подачу готовой продукции в горячем виде в короткий промежуток времени, ограниченный школьной переменной? Сколько понадобится одновременно таких установок?

Работа по конструированию, начатая в классе, иногда продолжается и дома. Дети, охваченные одной идеей или замыслом, группируются в конструкторские бюро. Внутри одного бюро появляется разделение труда по разработке проектов отдельных частей агрегата. Как правило, в таком временно собранном коллективе появляется лидер.

После домашней доработки и состыковки отдельных проектов в один начинается защита, которая представляет собой интересное зрелище, где оппонентами уже являются группы желающие доказать, что их проект лучше.

Ролевые игры — неотъемлемая часть нашей жизни. Игровые формы обучения — это учебный труд, освещенный радостью перевоплощения, перехода в мир, создаваемый творчеством.



КУДИЦ - ведущий учебно-методический
центр в области информатизации
образования

КУДИЦ представляет

уникальный программно-методический комплекс «Компьютеры в системах информации» (ПМК N5)

Вы познакомитесь с системой компьютерных телекоммуникаций и сможете представить учащимся работу телекоммуникационных сетей, а также сформируете элементарные навыки работы в среде электронной почты и телеконференций.

ПМК N5 является составной частью базового двухгодичного курса «Информатика» (см. «Информатика и образование» №3, 1993 г.), который с успехом работает более чем в 1400 школах России и стран СНГ.

В состав курса включены:

ПМК N1

Первые уроки информатики
Основы алгоритмизации

ПМК N2

Деловые применения ЭВМ

ПМК N3

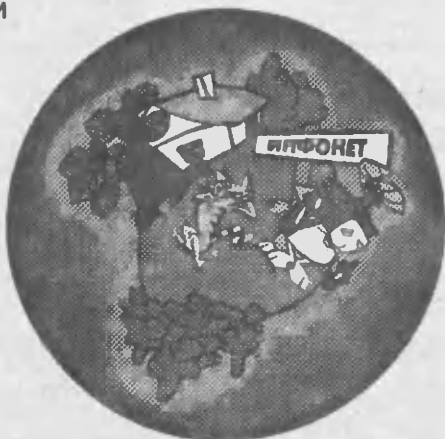
Элементы компьютерной графики

ПМК N4

Элементы компьютерного моделирования

ПМК N5

Компьютеры в системах передачи информации



Познайте мир с нами!

- Комплексные программы информатизации образования республики, края, области, города
- Современные компьютерные классы «под ключ»
- Программно-методические комплексы
- Обучение учителей, методистов и администраторов
- Телекоммуникационные образовательные проекты



107078, Москва,
Садовая-Черногрозская, 4

Тел.: (095) 207-07-09
Факс: (095) 207-08-08
E-mail: pilotNet@glas.apc.org

Познайте мир с нами!

В журнале «Информатика и образование» регулярно публикуются материалы, освещающие «Пилотный проект», который был реализован КУДИЦ. В 1994 г. КУДИЦ отмечает свой пятилетний юбилей. В связи с этим мы обратились к научному руководителю КУДИЦ, члену редколлегии журнала «Информатика и образование», кандидату технических наук Борису Григорьевичу КИСЕЛЕВУ, с которым вы уже встречались на страницах нашего журнала.

— Борис Григорьевич, что Вы считаете самым значимым в работе КУДИЦ за эти годы?

— Наиболее значительной работой и сейчас остается проект «Пилотные школы», успешно выполненный, несмотря на сжатые сроки (два года). Сегодня сотни школ работают, используя технику и программно-методические комплексы (ПМК), которые были разработаны и предоставлены школам в рамках этого проекта.

Именно в проекте «Пилотные школы» впервые был реализован комплексный подход к проблеме информатизации образования: поставка компьютеров высокого качества, подготовка педагогических кадров, разработка и поставка программно-методического обеспечения, методическое сопровождение реального учебного процесса. Кстати, косвенное свидетельство успеха этой работы — появление рекламы других организаций, которые в качестве аргумента для продвижения своих программных продуктов указывают на их связь с проектом «Пилотные школы». Имя КУДИЦ стало синонимом нового подхода к информатизации образования и в этом качестве вошло в название некоторых новых фирм.

— Проект «Пилотные школы» формально завершился в 1991 г. Какие разработки были сделаны за прошедшие два года?

— Изменилась страна, изменились условия работы и финансирования. Но сохранилось главное в работе КУДИЦ — ориентация на оказание помощи и поддержки учителю, который хочет использовать современные информационные технологии (ИТ) в обучении. Сохранились тесные связи со школами. Сохранилось ядро КУДИЦ — люди, которые начинали проект «Пилотные школы», закладывали основные идеи, реализованные в этом проекте. Проникновение современных ИТ в школы идет, по нашему мнению, по двум основным направлениям.

Первое — курс информатики, новая модель которого была разработана и апробирована в рамках проекта «Пилотные школы». Этот курс, в отличие от предыдущих, ориентирован на овладение современным компьютером как принципиально новым средством познания окружающего мира. Создана целая серия ПМК, в том числе по принципиально новым разделам курса информатики, ранее отсутствовавшим в средней школе. Среди них:

- ПМК «Деловые применения ЭВМ», разработанный при участии фирмы «Борланд» (в настоящее время им пользуются около 2 тыс. школ);

- ПМК по компьютерному моделированию. Благодаря этому ПМК методы работы с компьютером, ранее доступные только в высшей школе и в научных учреждениях, впервые пришли в среднюю школу;

- ПМК по телекоммуникациям, позволяющий детям и учителям узнать и практически освоить возможности современных телекоммуникаций, благодаря которым компьютер становится окном, открывающим доступ ко всему миру, к огромным богатствам накопленной в мире информации.

Второе направление — начальная школа. Мы издали ПМК «Роботландия», ориентированный на раннее изучение информатики, на развитие навыков логического мышления, проведения исследования с использованием ИТ. Дети, освоившие элементы ИТ в рамках курса «Роботландия», смогут активно использовать их при изучении других предметов.

— Что нового планируется в деятельности КУДИЦ в ближайшее время?

— В прошлом году была опубликована разработанная нами программа развития проекта «Пилотные школы» на 1993—1994 гг., сейчас мы ведем работу над программой на период 1995—2000 гг. В стратегии КУДИЦ появились две основные тенденции.

Первая. Ориентация не на типовую школу, а на стандарты образования. Это очень важно. Пожалуй, только современные ИТ позволяют гарантированно сохранить то качество образования, которым традиционно гордилась отечественная школа, при растущем разнообразии форм и методов обучения и видов учебных заведений.

Вторая. Ориентация на работу с регионами, которые сейчас занимают гораздо более активную позицию и стремятся вкладывать появившиеся у них материальные средства в развитие образования. КУДИЦ прилагает значительные усилия по оказанию регионам помощи, в первую очередь методической, связанной с разработкой и выполнением региональных проектов информатизации образования. Я хотел бы отметить региональные проекты в Сургуте, Калмыкии и ряд других.

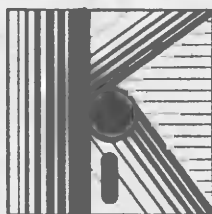
И последнее. Сейчас КУДИЦ расширяет свою деятельность, и нам необходимы дополнительные силы. Поэтому мы приглашаем методистов по различным предметам, а также учителей-практиков для работы по различным направлениям использования ИТ в современной школе. Звоните нам и пишите.

Борис Григорьевич, мы желаем Вам и коллективу КУДИЦ творческих успехов и надеемся, что один из номеров журнала в этом году будет посвящен юбилею фирмы. В нем мы расскажем



К Вашим услугам:

- ★ *Поставка компьютерных классов "под ключ" на базе IBM-совместимой техники различной конфигурации, УКНЦ, БК0011М;*
- ★ *Изготовление мебели для учебных заведений (парты, столы, стулья, скамейки, стеллажи, шкафы и др.)*



КОЛЛЕДЖ

Последние программные и методические разработки:

- ★ Программно-методический комплекс "Информатика" для УКНЦ, КУВТ-86, ДВК, IBM.
- ★ Пакет административных программ для УКНЦ ("Завуч", "Бухгалтерия", "Склад и реализация").
- ★ Информационно-поисковая система для УКНЦ, IBM "DATA-MANAGER" (база данных, электронные таблицы, текстовый редактор в едином интерфейсе).
- ★ Обучающие и моделирующие программы по общеобразовательным предметам (физика, экономика, русский язык).
- ★ Развивающие игровые программы для IBM.

Каталог программного обеспечения высылается бесплатно

Наш адрес: 107005, Москва, Волховский пер., д.11,
Фирма "Колледж"

Телефоны: 265-6265, 267-7058 Телефакс: 265-6265

Ю. А. Первин, *Переславль-Залесский*

УЧИТЕЛЬСКИЕ СЕМИНАРЫ ПО РОБОТЛАНДИИ

В рамках проекта «Пилотные школы» получила широкое распространение программно-методическая система «Роботландия», которая может рассматриваться как курс раннего обучения информатике (типовая ориентация курса — 3-4 классы общеобразовательной школы) или как база для интегрированного подхода к компьютеризации начального образования. В настоящее время накоплена достаточно обширная библиография по этой системе. Непосредственно для проекта «Пилотные школы» разработана серия методических пособий: учебник для школьника в восьми тетрадах, методические поурочные комментарии к курсу в семи тетрадах-разделах, сборник пользовательских (ориентированных на школьного учителя информатики) инструкций.

На основе этих материалов построена деятельность курсов повышения квалификации учителей или семинаров, которые ставят целью подготовить учителей к работе в школьном кабинете информатики, оснащенный РС-совместимыми компьютерами (с мониторами не ниже EGA или VGA).

Рекомендации, сложившиеся в ходе таких учительских семинаров, легли в основу предлагаемой статьи.

Цель и назначение курсов переподготовки учителей

Прежде чем приступить к преподаванию информатики младшим школьникам, учитель должен детально ознакомиться с содержанием курса и методикой обучения. Этому знакомству и посвящаются учительские семинары переподготовки.

В теоретическом аспекте учителю важно показать, что содержание обучения в школьном курсе информатики — это, прежде всего, видение информационной картины окружающего мира и роли информационных процессов в современном обществе. Такая мировоззренческая роль курса информатики является в значительной мере новой для нынешней школы, поэтому учителя, готовящегося к преподаванию нового курса, следует убедить в философской важности этой дисциплины и ее фундаментальности для гармоничного развития школьника.

В практическом аспекте слушателю семинара предстоит познакомиться с новыми программами, их возможностями, приемами управления, методикой обучения. В этом плане имеет особое значение практическая работа учителя за персональным компьютером.

Основными задачами, возникающими перед учителем при изучении Роботландии, являются:

- формирование навыков работы с устройствами ввода информации в персональный компьютер (клавиатура, мышь);
- ознакомление с простейшими командами операционной системы: просмотр (полный и по задаваемому шаблону) имен файлов, копирование информации, чтение и запись файлов, поиск файлов в системе каталогов, вывод информации на принтер;
- самостоятельное овладение системами подсказки — справочниками (иерархическими и контекстными);
- активное освоение программного материала при консультационном участии руководителя семинара, включая формирование конфигурации школьного кабинета;
- практическое выполнение всех (без исключения) лабораторных работ;

- самостоятельное составление учителем лабораторных работ и упражнений (формирование заданий);
- проверка закрытых (для учащихся) возможностей Роботландии (демонстрационные режимы Мудрого Крота, запрещение инструментов в Художнике и т. п.);
- знакомство с дополнительными возможностями системы (редактирование задач Мудрого Крота, координаты в Художнике, знаки альтерации в Шарманщике и т. п.);
- творческое участие в организуемых руководителем семинара дискуссиях по концептуальным вопросам Роботландии.

Структура методического сопровождения

В рамках деятельности курсов переподготовки учителя-слушатели могут пользоваться следующими методическими материалами:

- справочниками и справочными системами, встроенными в соответствующие программы Роботландии;
- учебником (книгой для чтения) для учащихся, отражающим основное содержание курса обучения. Работая с этой книгой, учитель должен проверить (на компьютере, письменно или устно) все задания, которые размещаются в конце большинства параграфов книги для чтения;
- пособием (книгой для учителя), содержащим методику курса. (Несмотря на рекомендательный характер пособия, учитель должен тем не менее внимательно с ним ознакомиться);
- сборником инструкций «Программные средства Роботландии», включающим в себя полное описание пользовательского интерфейса всех компьютерных программ курса;
- библиографией по системе Роботландия, содержащей опубликованные материалы для факультативного чтения.

В настоящее время готовится к печати книга «Лекции по Роботландии», которая построена по обсуждаемому здесь ординарному учебно-методическому плану.

Преподавателям и школьникам, интересующимся Роботландией, можно также порекомендовать недавно выпущенную издательством «Педагогика-Пресс» книгу А.А. Дуванова и Ю.А. Первина «Необыкновенные приключения Пети Кука в Роботландии». Эта популярная книга хотя и не претендует на роль учебника по курсу информатики, но может быть с успехом использована как материал для школьного кружка или для самостоятельного знакомства с основами информатики.

Требования к уровню подготовки учителей

Программно-методическая система Роботландия ориентирована на обучение учащихся начальной школы. Поэтому основные пользователи Роботландии — это учителя начальных классов. В настоящее время многие студенты педагогических вузов и педучилищ изучают информатику, и это существенно упрощает задачи организации курсов переподготовки учителей. Особое внимание при подготовке таких курсов обращается на контингент учителей, впервые приступающих к систематическому использованию компьютеров и основ информатики в их практической педагогической деятельности. Курс, ориентируемый на такую категорию слушателей, в дальнейшем называется *ординарным* и включает в себя 30 часов обязательных занятий, 13 лабораторных работ и 5 факультативных дискуссий. При этом предполагается, что предварительных знаний и практических навыков общения с компьютерами у этой категории слушателей нет.

Учитывая, что первоочередными объектами учебного применения программ Роботландии станут школы — участники проекта «Пилотные школы» и, следовательно, имеющие опыт практического использования компьютеров, важно предусмотреть вторую категорию слушателей, которых можно отнести к подготовленным пользователям. Такие слушатели знакомы с основными понятиями информатики, назначением отдельных устройств компьютера, простейшими командами операционной системы. Курс, ориентируемый на контингент подготовленных пользователей, называется *ускоренным*. Он состоит из 10 часов обязательных занятий и трех факультативных дискуссий.

Учителю, приступающему к занятиям на курсах переподготовки, предлагается самостоятельно определить для себя тип курса.

Структура курсовых занятий

Все обязательные занятия курсов (кроме, возможно, первого часа) проходят в кабинете, оборудованном персональными компьютерами типа PS/2. Состав группы обычно определяется количеством свободных машин с тем, чтобы каждый слушатель имел к ним постоянный доступ.

Основной учебный материал курса организован в виде отдельных модулей-лекций. Порядок тем-лекций соответствует, во-первых, рекомендациям учебно-тематического плана для семинаров повышения квалификации работников образования, а во-вторых, нумерации выпущенных проектом «Пилотные школы» книг, являющихся пособиями для учителя.

Наряду с лекциями в курс (план семинара повышения квалификации) включены два вида практических работ.

Один из них — лабораторные работы. Часть их дублирует те работы, которые рекомендованы для школьного курса Роботландии, так как учитель должен детально знать особенности своих предстоящих практических занятий в школе. Другая — позволяет учителю взглянуть на знакомые приемы в новых средах. Такого рода работы очень важны, так как показывают слушателю, насколько неформально усвоены им необходимые знания.

Второй вид практической работы — эпизодические компьютерные упражнения, названные здесь лабораторными включениями. Они предусматривают обращение слушателей к компьютерам в виде кратких прерываний лекции, инициируемых преподавателем.

Деление компьютерных упражнений на лабораторные работы и лабораторные включения может восприниматься как весьма условное, если пытаться увидеть в нем «большие» и «малые» упражнения. Естественнее первые считать упражнениями на закрепления полученных знаний, а вторые — иллюстрациями отдельных приемов работы с компьютером.

Наконец, еще один вид деятельности слушателей — это коллективные и, как правило, «бесмашинные» семинарские занятия. Эта форма деятельности названа здесь дискуссиями. Тем самым выражена надежда на то, что темы, предложенные для такого рода учебной работы, способны стать базой для обсуждения вопросов, по которым у слушателей может сформироваться неоднозначное мнение. Во всяком случае дискуссии предполагают высокий уровень самостоятельности слушателей как в подготовке вопросов для обсуждения, так и в генерировании актуальных тем подобных дискуссий.

Рекомендации, посвященные подготовке учителей ускоренного потока, определяют объем курса в зависимости от навыка слушателей, но не лимитируют организацию курса. Например, допустимыми считаются курсы изучения различных программных систем проекта «Пилотные школы» с добавлением занятий по Роботландии в той же мере, как курсы, посвященные исключительно Роботландии.

Так, например, возможна организация недельных (шестидневных) курсов по следующей схеме.

Первый день

Два часа Роботландии; другие дисциплины.

Второй день

Один час Роботландии; первая дискуссия; другие дисциплины.

Третий день

Два часа Роботландии; другие дисциплины.

Четвертый день

Два часа Роботландии; другие дисциплины.

Пятый день

Два часа Роботландии; вторая дискуссия; другие дисциплины.

Шестой день

Один час Роботландии; третья дискуссия; другие дисциплины; зачет.

Ускоренный курс может быть проведен в течение двух полных дней. Расписание занятий будет следующим.

Первый день

Шесть обязательных часов; факультативная дискуссия.

Второй день

Четыре обязательных часа; две факультативные дискуссии; зачет.

Для обычного курса также могут быть предложены две схемы. Одна из них — двухнедельная — рекомендуется при изучении Роботландии наряду с другими дисциплинами; другая — недельная — при изучении исключительно программно-методической системы Роботландии.

В домашних заданиях на курсах переподготовки нет необходимости, они факультативны и определяются ходом занятий и степенью усвоения материала.

Методические рекомендации по проведению курсов переподготовки учителей

Независимо от конфигурации кабинета, хотя бы на время обучения на курсах, каждый учитель-слушатель должен иметь диски с программами Роботландии. Работая с дисками, учитель-слушатель часть заданий выполняет самостоятельно, другую часть — под руководством или при консультационном участии преподавателя в ходе лабораторных работ. Для того чтобы учитель-слушатель мог непосредственно оценить время, необходимое для загрузки машин программами Роботландии, на курсовых занятиях эта процедура выполняется в учебное время. При этом преподавателю курсов весьма важно эффективно использовать загрузочные паузы.

Возможный вариант организации учебного времени состоит в том, что загрузка программ в учебное время выполняется лишь на первых двух занятиях. Далее слушатели, ощутившие влияние загрузочных пауз на организацию урока, соглашаются работать в «школьном» режиме: с загрузкой программ перед занятиями. В этом случае за ними сохраняется ответственность самостоятельной проверки (за рамками курса) длительности загрузочных пауз для каждой из программ Роботландии.

По мере возможности преподаватель заменяет свой монолог групповым обсуждением тех ситуаций, которые возникают на экранах слушателей в ходе выполнения лабораторных и практических работ. Активность слушателей, совмещаемая, по возможности, с эффективным использованием учебного времени, — основной методический принцип курсов переподготовки.

Работа с карандашом и тетрадь — неизбежная форма деятельности слушателей. Тем не менее, всегда, когда это оказывается более эффективно, слушатели должны использовать возможности компьютера, и в частности машинной обработки текстовой информации — ее хранения, упорядочения, поиска, печати.

Дискуссии, рекомендуемые в курсе, чрезвычайно полезны. Они позволяют слушателю закрепить полученные знания и связать конкретные практические навыки с общими теоретическими обоснованиями, а преподавателю — оценить уровень усвоения материала слушателями и их умение ориентироваться в новом предмете.

Желательно применять в курсе переподготовки те технические средства, которые упоминаются в учебнике и пособии. В частности, кодоскоп с фломастерами следует использовать всюду вместо доски с мелом. В тех случаях, когда пособие рекомендует обращение к демонстрационному экрану, а в классе большой экран отсутствует, руководитель семинара может использовать для этой цели экран центральной машины, развернув его в сторону слушателей.

Как известно, одной из забот школьного учителя является создание в классе подгрупп и закрепление детей за рабочими местами. Как правило, в школе эту задачу не удается решить оптимально: каждому ребенку — свой компьютер, свое индивидуальное рабочее место. Поэтому в

ходе нескольких лабораторных работ руководитель семинара может предложить слушателям объединиться в пары, работающие за одним компьютером с тем, чтобы слушатели смогли оценить ту психологическую обстановку, в которой окажутся их будущие ученики. Более того, рекомендуется несколько раз сменить пары слушателей, чтобы учителя могли проникнуться чувством ответственности за подбор пар учащихся по принципу психологической совместимости.

В приводимых ниже двух вариантах курса учебных программ основной упор сделан не только на методику обучения информатике, но и на изучение программных и технических средств. Это связано с тем, что слушателем является, как правило, опытный педагог, которому сравнительно нетрудно изучить вопросы методики самостоятельно при наличии подробного пособия. Практика же общения с компьютером или, во всяком случае, новыми компьютерными программами у школьных учителей обычно невелика. Эти обстоятельства были учтены при разработке учебных программ курса.

Программа курса переподготовки учителей ускоренного потока (10 часов + 3 дискуссии)

Первый час

Основные концепции системы раннего обучения информатики младших школьников. Структура курса. Особенности методики раннего обучения информатике.

Второй час

Структура программных средств Роботландии. Соотношение методической и программной частей системы. Монитор. Запуск монитора. Настройка монитора на конфигурацию кабинета.

Третий час

Формирование простейших навыков общения с компьютером. Тренажеры (Меню, Курсор). Программа Правилка. Справочник. Формирование вспомогательных файлов. Программы Привет, Угадайка.

Четвертый час

Алгоритмические этюды (Перевозчик, Монах, Конюх, Переливашка). Протокол. Откатка. Информационная система «Устройство компьютера». Методические вопросы алгоритмического направления курса.

Пятый час

Исполнители (Квадратик, Автомат, Машинист). Система команд. Общая схема знакомства с исполнителем. Черные ящики. Буквоед. Методика уроков, посвященных исполнителям и черным ящикам.

Шестой час

Текстовый редактор Микрон. Операции со строками. Копирование. Файлы. Имена файлов. Лабораторные работы Микрона. Методика обучения приемам текстовой обработки информации.

Седьмой час

Графический редактор. Основные области экрана. Инструментарий. Лабораторные работы Художника. Образовательные и воспитательные аспекты редактирования графической информации.

Восьмой час

Файлы (чтение, запись, каталог). Три типа файлов — картинки, фрагменты, объекты. Путь в каталоге. Понятие конфигурации задачи. Методика изучения понятий, связанных с файловыми системами.

Девятый час

Работа с музыкальной информацией в музыкальном редакторе Шарманщик. Образовательные и воспитательные аспекты редактирования музыкальной информации.

Десятый час

Программирование на языке исполнителя Кукарача. Управляющие структуры: ветвления и циклы. Программа. Запись программы и средств. Редактирование программ. Методика обучения программированию.

**Программа курса подготовки учителей ординарного потока
(30 часов + 13 лабораторных работ + 5 дискуссий)**

Вводное занятие

Представление курса, цели, структура учебного плана, организация учебного процесса.

Первый час

Компьютеризация образования — социальный заказ информационного общества.

Второй час

Концепции Роботландии. Основные педагогические направления курса раннего обучения информатике.

Третий час

Методические компоненты Роботландии. Структура пакета методических материалов. Взаимоотношения учебника (книги для чтения) для ученика и пособия для учителя. Планирование лабораторных работ.

Четвертый час

Введение в файловую систему персонального компьютера. Первое представление об операционной системе. Выбор и запуск программы. Понятие файла. Имя файла, полное имя, расширение. Каталоги и подкаталоги. Древесная структура файловой системы.

Пятый час

Конфигурация программных средств Роботландии. Комплект поставки. Монитор. Тематические разделы. Настройка на конфигурацию. Вызов программ системы.

Шестой час

Первые клавиатурные тренажеры. Тренажер Меню. Программа Курсор. Режимы программы Курсор. Демонстрационный режим. Настройка времени в программе Курсор.

Лабораторная работа «Демонстрационная версия Роботландии».

Лабораторная работа «Корректировка констант времени в программе Курсор».

Седьмой час

Мудрый Крот. Понятие уровня задачи. Открытые и закрытые демонстрационные режимы в программе Мудрый Крот. Представление о редакторе лабиринтных задач.

Лабораторная работа «Решение задач Мудрого Крота».

Лабораторная работа «Редактирование задач Мудрого Крота».

Восьмой час

Корректировка ошибок набора. Методика начальных этапов работы с клавиатурой. Тренажер Правилка. Режимы корректировки. Понятие контекстного справочника.

Лабораторная работа «Правилки. Формирование тестовых файлов для Правилки».

Девятый час

Структура информационной линии курса. Определения и описания понятий. Виды информации. Типы информационных процессов.

Десятый час

Информационная схема ЭВМ. Компьютер как информационная машина. Программа Компьютер. Мышь. Соответствия кнопок и клавиш. Методика проведения уроков с программой Компьютер.

Одиннадцатый час

Алгоритмические этюды. Общие свойства программ-этюдов. Протокол. Откатка. Игровая форма урока. Программы Перевозчик, Конюх, Переливашка.

Двенадцатый час

Рекурсия. Программа Монах. Понятие рекурсии. Урок по теме рекурсия с программой Монах. Буквенные обозначения алгоритмов.

Тринадцатый час

Случайные события в учебных программах. Программы Привет и Угадайка. Стратегия поиска. Бинарный поиск. Программа Морской бой.

Четырнадцатый час

Исполнитель и его память. Программа Машинист. Задачи для Машиниста. Понятие стека. Структуры памяти. Организация хранения информации.
Лабораторная работа «Машинист бронепоезда».
Лабораторная работа «Обходчик».

Пятнадцатый час

Общая схема знакомства с исполнителем. Исполнители. Понятие команды. Система команд исполнителя. Исполнитель Квадратик. Арифметические исполнители. Исполнитель Автомат.

Шестнадцатый час

Модель процессора. Исполнитель Плюсик. Арифметические задачи Плюсика. Длинный и короткий стеки. Уроки математики с исполнителем Плюсик. Разделение функций исполнителя и компьютера.

Семнадцатый час

Анализ алгоритмов. Игры в «черные ящики». Буквояд, его режимы и алгоритмы; Самостоятельные работы и соревнования с Буквоядом на уроке.

Первая дискуссия: «Игровая форма урока как пропедевтика компьютерных занятий и лабораторных работ в курсе раннего обучения информатике младших школьников».

Восемнадцатый час

Редактирование информации. Текстовый редактор Микрон. Понятия форматирования абзацев и шаблона строки. Операции со строками. Справочник Микрона.
Лабораторная работа «Приемы обработки текстовой информации».

Девятнадцатый час

Дисковые операции редактора. Копирование информации в текстовом редакторе. Запись и сохранение на диске. Файл, структура имени файла. Чтение, каталог. Методика обмена текстами.

Двадцатый час

Организация практических работ с текстовым редактором. Лабораторные работы с текстами.

Диктанты и грамматические упражнения. Проверка правильности. Печать текстов. Стенгазета и компьютерная газета.

Лабораторная работа «Упражнения с Микроном».

Двадцать первый час

Графический редактор Художник. Цвет. Палитра. Шаблон цветового заполнения. Инструменты. Режимы. Конфигурация редактора и инициализация задач.

Лабораторная работа «Инструментарий графического редактора».

Двадцать второй час

Файлы графической информации. Имена и расширения имен. Запись и сохранение файла на диске. Три типа файлов: картинки, фрагменты, объекты. Копирования и перемещения.

Двадцать третий час

Файловая система редактора. Каталоги и подкаталоги. Путь в системе каталогов. Области применения трех типов файлов.

Лабораторная работа «Коллажи картинок, фрагментов, объектов».

Двадцать четвертый час

Музыкальная информация и музыкальный редактор Шарманщик. Система музыкальных понятий, используемых в курсе раннего обучения информатике. Набор на клавиатуре простых мелодий. Требования к учителю.

Двадцать пятый час

Дополнительные возможности Шарманщика. Знаки альтерации, знаки при ключе, точка, паузы. Файловая система музыкального редактора. Методические аспекты использования Шарманщика.

Лабораторная работа «Концерт».

Вторая дискуссия: «Возможности программно-методической системы Роботландия как средства организации междисциплинарных связей».

Третья дискуссия: «Прикладная направленность уроков по темам, связанным с редактированием информации (стенгазеты, вернисажи, концерты)».

Четвертая дискуссия: «Общие принципы редактирования информации в редакторах Роботландия как база новых информационных технологий обучения».

Двадцать шестой час

Модель компьютерного управления исполнителем. Программа Кукарача как сложный исполнитель. Система команд Кукарачи. Разделение функций исполнителя и компьютера.

Двадцать седьмой час

Редактирование программ. Программа. Описание и выполнение программ. Принципы редактирования программ. Рабочее поле программы. Программа и среда. Файлы Кукарачи.

Двадцать восьмой час

Управляющие структуры. Перевернутые кубики Кукарачи. Ветвления: полные и сокращенные формы. Два вида повторения. Рекурсии.

Лабораторная работа «Задачи для Кукарачи»

Двадцать девятый час

Формы учебной работы с программно-методической системой Роботландия. Система индивидуального проектирования. Урок, консультация, кружок, летний лагерь, заочная школа.

Пятая дискуссия: «Программные средства Роботландия как инструмент творческой деятельности учащихся в системе проектов».

Тридцатый час

Будущее Роботландии. Перспективы развития Роботландии: интегрированные системы, коммуникации, аппаратные средства, параллельные исполнители.

**Список тем для дискуссий
в ускоренном курсе переподготовки учителей**

Дискуссия первая: «Сравнительная оценка методических возможностей системы программных исполнителей и учебно-ориентированных языковых систем программирования в курсе информатики для младших школьников».

Вопросы для дискуссии:

- требования к учебным языкам программирования;
- критика бесструктурных языков (Бейсик);
- особенности языковых программных систем, ориентируемых на младших школьников (ЛОГО);
- многообразие учебных сред как дидактический принцип, обосновывающий актуальность систем исполнителей;
- практический опыт использования систем программных исполнителей в отечественных разработках учебного программного обеспечения;

Дискуссия вторая: «Возможности программно-методической системы Роботландия как средства организации междпредметных связей».

Вопросы для дискуссии:

- программные средства обработки текстов на уроках русского и иностранного языков (диктанты, грамматические упражнения);
- возможности текстового редактора на уроках физики, химии, математики (работа с формулами);
- арифметические исполнители и их роль в математическом образовании младших школьников;
- графический редактор и его роль в математике (геометрические фигуры, координаты);
- графический редактор на уроках рисования и черчения;
- контурные карты по истории и географии на экране графического редактора;
- музыкальный редактор на уроках музыки и пения в общеобразовательной школе;
- взгляд на Роботландию как на интегрированный курс начального образования, базирующийся на фундаментальных понятиях информатики.

Дискуссия третья: «Программные средства Роботландии как инструмент творческой деятельности школьников в системе проектов».

Вопросы для дискуссии:

- эстетические принципы оформления текстов средствами адаптированного учебного текстового редактора;
- технология подготовки детских книжек и стенгазеты;
- пути использования графического редактора в художественном воспитании школьников (уроки рисования в общеобразовательной школе, кружки машинной графики в школе, дополнительные занятия по машинной графике в детской художественной школе);
- оценка объема музыкальных понятий, необходимых в интегрированном курсе начального образования и вводимых средствами музыкального редактора;
- воспитательные аспекты музыкального редактора: подготовка концерта, обсуждение форм музыкальных произведений, рассказ о творчестве ведущих детских композито-

ров, интернациональное воспитание, использование в репертуаре концерта — фольклор разных народов.

Список тем для дискуссий в ординарном курсе переподготовки учителей

Дискуссия первая: «Игровая форма урока как пропедевтика компьютерных занятий и лабораторных работ в курсе раннего обучения информатике младших школьников».

Вопросы для дискуссии:

- многоэтапная методика знакомства с алгоритмами и исполнителями: игра в классе, игра дома, игровая модель на компьютере;
- организация игр-спектаклей, предшествующих компьютерной лабораторной работе (Перевозчик, Машинист, Конюх);
- организация игр-соревнований для всего класса (турнир «черных ящиков»);
- игры на уроке (Мудрый Крот, Морской бой);
- Игры в курсе информатики в свете возрастных особенностей учащихся.

Дискуссия вторая: «Возможности программно-методической системы Роботландия».

Дискуссия третья: «Программные средства Роботландии как инструмент творческой деятельности школьников в системе проектов».

Дискуссия четвертая: «Прикладная направленность уроков по темам, связанным с редактированием информации (стенгазеты, вернисажи, концерты)».

Вопросы для дискуссии:

- актуальные задачи учебной и внешкольной деятельности как область использования редакторов Роботландии;
- технология выпуска стенгазеты;
- технология издания брошюры;
- вернисажи на школьных уроках (динамика заданий: фиксированные задачи;
- упражнения на применение конкретного инструмента;
- рисование на свободную тему);
- принципы распределения заданий при подготовке к компьютерному концерту.

Дискуссия пятая: «Общие принципы редактирования информации в редакторах Роботландии как база новых информационных технологий обучения».

Вопросы для дискуссии:

- редактирование информации как один из методических инструментов в обучении принципам проектирования;
- копирование информации как принцип автоматизации рутинной работы. Общие приемы редактирования и специфические особенности отдельных редакторов;
- работа с внешними носителями — одна из основных задач информационного редактора. Особенности чтения и записи информации в каждом редакторе;
- организация каталогов в редакторах. Общие приемы поиска информации в каталоге. Древесная структура в каталоге. Отличия возможностей разных редакторов.

М. В. Сякина, Ю. А. Первин

ИСПОЛНИТЕЛЬ ТЯНИ-ТОЛКАЙ НА УРОКАХ РОБОТЛАНДИИ

Продолжая серию публикаций по методике раннего обучения информатике с помощью программно-методической системы Роботландия, эта статья раскрывает тему «Коммуникации в локальной сети», не вошедшую в комплект товарной поставки системы. Описываемое программное обеспечение темы реализовано для локальной сети Ямахи MSX-1.

Вместе с тем, компьютерные коммуникации настолько важны в приобщении современных детей к технологиям нового информационного общества, что авторам представляется актуальным обсуждение педагогического эксперимента по этой теме.

Уроки по теме «Коммуникации» с применением исполнителя Тяни-Толкай были проведены в школах № 6 г. Переславля-Залесского и № 64 и 76 г. Ярославля. Программное обеспечение темы предусматривает телекоммуникации школьников. Однако отсутствие модемов в экспериментальных школах ограничило эксперимент уроками в стенах одной классной комнаты. Некоторое приближение к идеальным условиям компьютерной коммуникации обеспечивалось тем, что дети предварительно не получали сведений о своем напарнике по заданию.

Общие замечания по теме

Одной из целей введения в курс раннего обучения информатике программы «Тяни-Толкай» явилось создание условий, которые позволяют сформировать у детей соответствующие мотивы общения и организовать совместную деятельность учащихся. Организация совместной деятельности дает возможность сохранить взаимодействие между учащимися в учебном процессе, сформировать и развить у ребенка технику общения.

Работа с программой заключается в управлении исполнителем, при этом все компьютеры соединены попарно. Поэтому исполнителем управляют два человека (одновременно в локальной сети класса могут действовать несколько исполнителей).

Задача, которую ставит Тяни-Толкай, состоит в том, чтобы собрать слово из разбросанных на экране букв. Так как буквы находятся в различных частях экрана, необходимо правильно описать траекторию их движения. Программа имеет четыре серии задач, которые отличаются друг от друга уровнем сложности и рассчитаны на четыре занятия. Каждая серия имеет пять различных заданий.

В первой серии задач траекторию движения указывает машина пунктирной линией. Буква перемещается на отведенное ей место после того, как будет полностью описана траектория.

Во второй серии ученики должны сначала сами нарисовать траекторию. Машина показывает букву, которую надо двигать, а ученики с помощью клавиш-стрелок указывают путь. После того как траектория полностью начерчена, партнеры сравнивают свои траектории и выбирают одну из двух. Затем производится перемещение буквы.

В третьей серии задач перемещение букв происходит без предварительного построения траектории. Буква передвигается на один шаг после подтверждения партнером команды.

В четвертой серии задач на поле размещены препятствия, которые при перемещении буквы необходимо обойти.

Учитель со своей машины управляет работой класса: соединяет машины в пары, посылает задания, следит за ходом выполнения работы.

Ниже приводятся материалы по теме «Коммуникации» из учебного пособия для учащихся, разработанного для изучения этой темы.

Коммуникация

§ 1. Как можно играть со словами

Петя Кук и его верный друг Вася не знали, чем заняться. Они сидели на скамейке в сквере и читали слова задом наперед:

магазин — низагам

сквер — ревокс

небо — обен

компьютер — ретьюпмок

Со словом «компьютер» они возились долго и вдруг заметили, что по направлению к школе идет Сан Саныч.

— Что он будет делать в школе в воскресенье? — спросил Петя.

— Давай пойдем за ним, может, и для нас найдется занятие, — отозвался Василий.

Ребята догнали учителя, который шел, чтобы поработать на компьютере. Сан Саныч любит работать по воскресеньям, когда ему никто не мешает.

— Не найдется ли для нас занятие? — спросил Петя, когда они догнали Сан Саныча.

— А то мы со скуки читаем слова задом наперед. Например, Сан Саныч — Нас Чынас.

— Кажется, у меня есть подходящее для вас занятие, раз вы любите играть со словами, — усмехнулся Сан Саныч. — Это новый исполнитель — Тяни-Толкай.

— Кого тяни, кого толкай? — засмеялся Петя.

— Хотите посмотреть? — спросил Сан Саныч.

И вся компания направилась в кабинет информатики.

— У Корнея Ивановича Чуковского в сказке «Доктор Айболит» живет Тяни-Толкай с двумя головами. Сан Саныч, а ваш Тяни-Толкай тоже с двумя головами? — спросил Вася, когда они поднимались по лестнице.

— Да, но он живет в Роботландии, поэтому у него вместо голов два компьютера.

— Компьютерный Тяни-Толкай... — подумав произнес Петя.

Войдя в класс, ребята сели к своим компьютерам, а Сан Саныч занялся Тяни-Толкаем. Он нажал на какие-то клавиши, и на экранах у ребят появилось задание.

— Вы должны сообща собрать слово из этих букв, но прошу не разговаривать вслух, я буду работать, — сказал Сан Саныч.

— Как же мы будем работать сообща, если мой компьютер стоит здесь, а Петькин в конце класса, да еще разговаривать нельзя? — удивился Вася.

На это Сан Саныч спокойно ответил:

— Можно работать сообща, если компьютеры стоят в разных классах или даже в разных городах. Для этого только необходимо соединить их в коммуникационную сеть.

— Все понятно. Но не понятно, что за штука эта коммуникационная сеть, — сказал Петя.

— Слово «коммуникация» означает «связь». Дальше не сложно догадаться. Коммуникационная сеть — это компьютерная сеть, в которой можно общаться, передавая друг другу информацию, так как машины связаны друг с другом. Не говоря вслух, вы можете разговаривать с помощью компьютера.

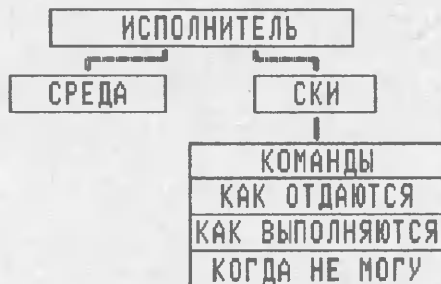
— Попробуем, — серьезно сказал Петя.

Сан Саныч познакомил ребят с исполнителем, и работа закипела.

Вопросы:

1. Что означает слово «коммуникация»?

2. Что такое коммуникационная сеть?



§ 2. Исполнитель Тяни-Толкай

СРЕДА

Посмотрите на рис. 1. Среда Тяни-Толкай — клетчатое поле с выделенной на нем площадкой. В некоторых клеточках поля расставлены буквы.

Ниже поля расположено окно, в котором записываются команды для исполнителя.

Тяни-Толкай работает в режиме коммуникационной сети. Работа ведется на паре компьютеров одновременно.

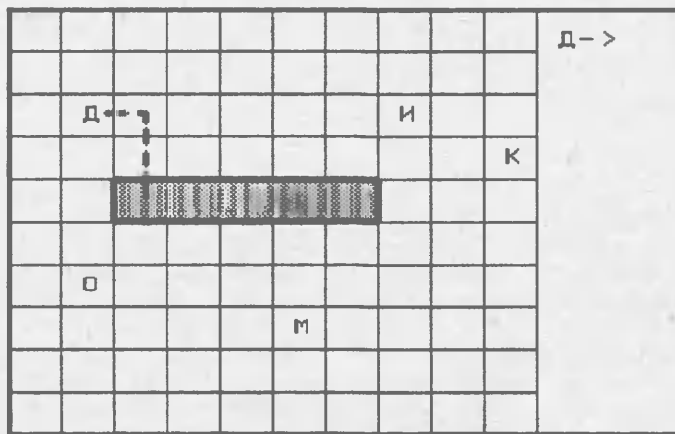


Рис. 1

КОМАНДЫ

Существует два вида команд:

команды перемещения;

команды подтверждения:

- «согласен с партнером,
- «не согласен с партнером.

КАК ОТДАЮТСЯ КОМАНДЫ

Команда перемещения составляется из перемещаемой буквы (заглавной русской) и стрелки. Например:

Д → О ←

Команда завершается нажатием клавиши выполнения.

Команды подтверждения отдаются клавишами «+» («согласен с партнером») и «-» («не согласен с партнером»).

КАК ВЫПОЛНЯЮТСЯ

А → — запоминает направление перемещения и букву, которая должна быть перемещена.

«согласен с партнером» — подтверждает действия партнера и исполняет последнее запомненное предписание;

«не согласен с партнером» — отменяет (удаляет из памяти) запомненное предписание.

Когда НЕ МОГУ

Ситуация НЕ МОГУ возникает, когда был сделан неверный ход (выход за пределы поля, уход с заданной траектории, перемещение внутри выделенной площадки).

Вопросы:

1. Как отмечается согласие или несогласие с партнером?
2. Какие команды не может выполнить Тяни-Толкай?

§ 3. Тяни-Толкай играет в слова

«Ходить по клеточкам несложно», — подумал Петя и сделал первый ход:

Д →

Затем нажал на клавишу выполнения, и... Тяни-Толкай ответил:

ЖДИТЕ ОТВЕТА

На экране у Васи в этот же момент появился записанный Петей ход и сообщение:

ВВЕДИТЕ ПОДТВЕРЖДЕНИЕ

«Петяка сделал правильный ход. Чего тут думать!» — и Вася нажал на клавишу «+».

Тяни-Толкай ему ответил:

ВЫ СОГЛАСНЫ

Компьютер запомнил этот ход и сказал Васе:

ВАШ ХОД

Тогда Вася написал:

Д ↓

и нажал клавишу выполнения.

На это Тяни-Толкай сказал:

ЖДИТЕ ОТВЕТА

и передал Васин ход Пете.

В этот момент Петя считал, сколько им надо сделать шагов, чтобы поставить букву на место. Он, не задумываясь, подтвердил ход Васи и сделал свой:

Д →

Когда Вася у себя на экране увидел, что хочет сделать Петя, он очень удивился. Он подумал: «Как это Петька собирается передвинуть букву в сторону, когда надо двигать вниз?!». Он нажал на клавишу «←». Тяни-Толкай ответил:

ВЫ НЕ СОГЛАСНЫ

и стер с обоих экранов этот ход. Следующий ход он предложил сделать Васе.

Ребята постепенно описали всю траекторию движения буквы, и она встала на свое место.

Тяни-Толкай нарисовал траекторию для следующей буквы.

Наконец, на свое место была поставлена последняя буква.

— Всё! — в один голос сказали ребята.

Вопросы и задания:

1. Какую линию можно назвать кратчайшей траекторией движения буквы?
2. Подсчитайте количество кратчайших траекторий для каждой буквы по рис. 2.

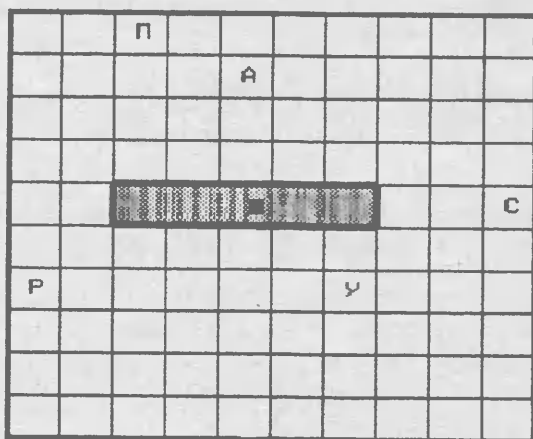


Рис. 2

3. Придумайте слова, из которых путем перестановки букв можно получить новое слово. (Такие слова называют анаграммами.)

§ 4. Выбор

В воскресенье класс Пети Кука собирался пойти в поход на озеро. Решили сделать стоянку у деревни Бережки. А Пете и Кате поручили составить маршрут движения. Ребята взяли карту и увидели, что в деревню Бережки ведут две дороги (рис. 3).



Рис. 3

Одна идет через деревню Дорожки, другая — по берегу озера.

— Идем через деревню Дорожки, — сказал Петя.

— Нет, надо идти вдоль берега озера, — ответила Катя.

— Зачем вдоль берега? — не соглашался Петя. — Дорога через деревню Дорожки — прямая и ровная. Мы не собьемся с пути.

— Зато дорога по берегу озера короче, — не уступала Катя.

Петя согласился с Катей. Решено было идти по берегу озера.

Этот поход всем очень понравился. Ребята вернулись усталые, но довольные.

В понедельник на уроке информатики ребята работали с исполнителем Тяни-Толкай. Сан Саныч отметил, что сначала необходимо начертить траекторию движения буквы. И она должна быть кратчайшей.

— Это похоже на то, как мы с тобой выбирали маршрут, — шепнул Петя своей соседке Кате.

— Траекторию надо чертить клавишами-стрелками, — объяснил учитель. — Если вы хотите стереть какую-то часть траектории, вернитесь назад по своим следам. После того как начертите всю траекторию, нажмите на клавишу выполнения.

— Буква встанет на свое место? — спросил нетерпеливый Петя.

— Нет, — ответил Сан Саныч. — На экране появятся две траектории. Одна — твоя, другая — твоего партнера. Необходимо выбрать одну (лучшую), т. е. кратчайшую. Для этого необходимо нажать «+» или «-», что означает «согласен» или «не согласен», соответственно.

— А что делать дальше? — спросил Саша.

— Дальше надо описать траекторию движения, как мы это делали на прошлом занятии, — ответил Сан Саныч. — Если вы не согласны друг с другом, то машина предложит вам нарисовать траекторию заново.

В конце урока, когда подвели итоги занятия, выяснилось, что те пары, которые часто не соглашались друг с другом, дольше выполняли задание.

Вопросы и задания:

1. Как чертится траектория?
2. Как выбирают наилучшую траекторию?
3. Придумать как можно больше слов, используя буквы: *а, о, е, п, р, с, ш, т* (каждую букву можно использовать только один раз.)

Методические рекомендации для учителя по изучению темы «Коммуникации» с использованием программы «Тяни-Толкай»

Урок 1

Тема урока: Компьютер как средство коммуникации.

Цели: формирование у школьников представления о компьютерных сетях; обучение взаимодействию в парах при работе на компьютере.

Программное обеспечение: исполнитель Тяни-Толкай.

План урока:

1. Проводится беседа о средствах коммуникации, предлагается провести такую игру: класс делится на группы по четыре человека; каждый ребенок получает коробочку с несколькими предметами. Эти предметы являются половинками некоторой фигуры (объемной или плоской). Вторая половинка находится обязательно у кого-нибудь из этой же группы. Свои предметы ребята не показывают друг другу. Игра заключается в том, чтобы определить, у кого находится вторая половинка. Для этого дети должны подробно описать свои предметы, рассказывая о них, и по словесному описанию попытаться определить у кого находится вторая половинка. Как только дети нашли своего предполагаемого партнера, они договариваются и одновременно выкладывают на стол свои фигурки, чтобы проверить правильность своего выбора. Побеждает та команда, у которой больше пар совпало.

Такая игра активизирует словарный запас ребенка, развивает пространственное воображение. Дети с удовольствием играют в нее, придумывая оригинальные описания. Игра может повторяться на нескольких занятиях.

2. Компьютер как средство общения.

Учитель рассказывает детям, что с изобретением компьютера появилось новое средство коммуникации, объясняя, что компьютер можно использовать не только как средство хранения и обработки информации, но и как средство передачи этой информации.

Необходимо объяснить, что означает понятие «коммуникационная сеть» (локальная и глобальная), объясняя это на доступных детям примерах.

3. Знакомство с исполнителем Тяни-Толкай.

Материал этой части урока совпадает с § 2 учебного текста.

4. Работа на компьютере: исполнитель Тяни-Толкай.

5. Обсуждение домашнего задания.

На дом задаются § 1, 2 и 3. Необходимо ответить на вопросы и выполнить задания в конце § 3 (дополнительные пояснения требуются к задачам 2 и 3).

Задаче 2 должно предшествовать обсуждение по примерному плану:

1) Какое слово должно получиться?

2) Начертите кратчайшую траекторию движения первой буквы.

3) Является ли эта траектория единственной?

4) Какие еще существуют кратчайшие траектории?

К задаче 3 достаточно привести примеры слов-анаграмм:

нос — сон

трос — сорт — рост — торс

исток — косит

Урок 2

Тема урока: Взаимодействие при решении задач на компьютере.

Цели: закрепление навыка работы за компьютером в паре, обучение нахождению оптимального варианта решения задач.

Программное обеспечение: исполнитель Тяни-Толкай.

План урока:

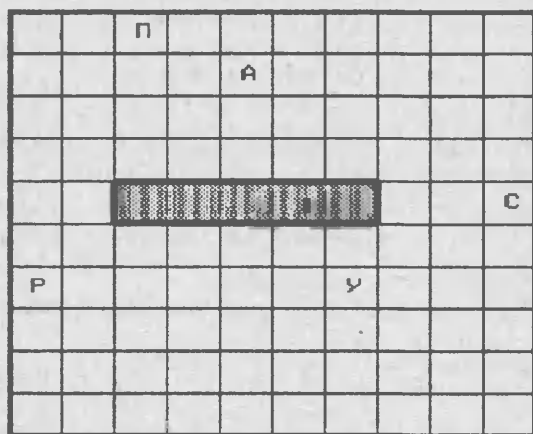
1. Анализ выполнения домашнего задания.

Анализ и проверка домашнего задания начинается с обсуждения вопросов:

1) Что такое коммуникационная сеть?

2) Что называется траекторией движения буквы?

- 3) Какие клавиши нажимают в случае согласия с партнером?
 4) Какие клавиши необходимо нажать в случае несогласия с партнером?
 Далее следует обсуждение задачи 2 (рис. 4).



Количество кратчайших траекторий для букв:

- П — 1,
 А — 3,
 Р — 5,
 У — 2,
 С — 4.

Рис. 4

Примерные вопросы:

- 1) Какое слово должно получиться в ходе решения задачи?
- 2) Начертите кратчайшую траекторию движения первой буквы?
- 3) Является ли эта траектория единственной?
- 4) Какие еще существуют кратчайшие траектории?

Учитель, записав это задание на пленке кодоскопа, при обсуждении решений наносит на чертеж все траектории.

При проверке задачи 3 можно организовать соревнование, оценивая по таким параметрам:

- кто больше придумал слов-анаграмм;
- кто придумал больше слов из букв одного слова;
- кто придумал самое длинное слово.

2. Упражнения с использованием Тяни-Толкая.

Учитель напоминает о правилах, в соответствии с которыми происходит дальнейшее выполнение работы, и вносит изменение в задание. В отличие от предыдущего урока ученик должен сам начертить траекторию движения. Этот урок является самым сложным из всего цикла занятий с Тяни-Толкаем, так как у детей могут возникнуть трудности с построением траектории. Поэтому рекомендуется более подробно объяснить сам процесс построения (он описан в § 4). Ниже описывается выполнение задания.

При получении задания на экране появляется изображение поля, на котором разбросаны буквы, составляющие заданное слово (буква, которой предстоит перемещение, выделена). После того, как траектория построена, она сопоставляется с вариантом партнера. Траектории могут совпадать или различаться. В первом случае учащиеся непосредственно приступают к сборке слова по ранее объявленным правилам. Тем, у кого траектории не совпали, предлагается сделать выбор. Право выбора первым предоставляется тому, у кого траектория мерцает. Для этого необходимо нажать на клавиши «+» или «-», обозначающие «согласен» или «не согласен», соответственно, а потом — клавишу выполнения. Затем право выбора предоставляется партнеру. Выбор производится так же, как и в предыдущем случае. После этого учащиеся приступают к выполнению задания.

Аналогичная работа проводится при перемещении каждой буквы. Задание считается выполненным, если собрано все слово.

3. Обсуждение домашнего задания.

4. Необходимо ответить на вопросы, помещенные в конце параграфа, и выполнить задания.

Урок 3

Тема урока: Компьютер как средство коммуникации.

Цели: обучение взаимодействию в парах при работе на компьютере; обучение нахождению оптимального варианта решения задач.

Программное обеспечение: исполнитель Тяни-Толкай.

План урока:

1. Анализ выполнения домашнего задания.

Обсуждение вопросов домашнего задания одновременно является повторением перед лабораторной работой. Поэтому стоит подробно разобрать следующие вопросы:

1) Какая траектория называется кратчайшей?

2) Как чертится траектория?

3) Как выбрать траекторию?

Проверку задачи можно провести в форме соревнования.

Критерии оценки могут быть различными: кто больше придумал слов, кто придумал самое длинное слово и т. д.

2. Работа на компьютере: исполнитель Тяни-Толкай.

На этом занятии используется третья серия задач.

Учитель должен уделить особое внимание тому, как дети соединены в пары. У каждого ребенка есть собственное решение этой задачи. Так как оно не единственное, то исход игры не ясен. Раньше закончит игру та пара, которая быстрее сумеет договориться. Так как дети не связаны рамками траектории, то после каждого сделанного хода могут появиться новые варианты ходов. Ребенок видит свой ход и ход своего партнера. Он должен выбрать лучший. При этом ходы могут быть равноценными, и может возникнуть спорная ситуация. То, как дети сумеют выйти из нее, определит исход игры. Школьники, не обладающие психологическими характеристиками лидера, как правило, договариваются быстрее. Если в паре соединены два лидера, может оказаться, что эта пара закончит задание последней.

Выполнение задания начинается с того, что ученики определяют, какое слово должно получиться. Некоторым детям иногда трудно это сделать, поэтому перед выполнением работы на компьютере предлагается провести следующую игру.

Водящий загадывает слово из трех неповторяющихся букв. Ребята по очереди говорят ему трехбуквенные слова. Водящий отвечает, сколько букв из этого слова есть в загаданном им слове. Такой диалог продолжается, пока слово не будет отгадано. Для наглядности рекомендуется записывать на пленке или на бумаге слова и соответствующее им количество букв. Например, водящий загадал слово «лес»; учащиеся ему говорят:

сон — ответ: 1 буква,

бор — ответ: 0 букв; это значит, что из алфавита можно вычеркнуть буквы: «б», «о», «р»,

сев — ответ: 2 буквы,

лак — ответ: 1 буква, и т.д.

Анализируя ответы водящего, ученики должны отгадать слово.

Эта игра необычна (является «словесным» аналогом известной игры с числами «Быки и коровы» — *Прим. ред.*) и поначалу сложна для детей. Однако она развивает логическое мышление ребенка, увеличивает его словарный запас. По окончании игры задание выполняется на компьютере.

3. Обсуждение домашнего задания.

Задача: из слова «коммуникация» придумать максимально возможное количество слов.

Урок 4

Тема урока: Решение коммуникационных задач на компьютере.

Цели: закрепление навыков взаимодействия в парах при работе на компьютере.

Программное обеспечение: исполнитель Тяни-Толкай.

План урока:

1. Анализ выполнения домашнего задания.

Учитель вместе с классом разбирает слова, придуманные детьми.

После разбора домашнего задания можно повторить игру, проводившуюся на прошлом занятии, используя слова, состоящие из четырех и пяти букв.

2. Работа на компьютере: исполнитель Тяни-Толкай.

На этом занятии выполняется четвертая серия задач.

Необходимо соединить детей в пары таким образом, чтобы избежать конфликтных ситуаций (если они возникали на прошлом занятии).

Как и на предыдущем занятии, выполнение задания начинается с определения слова. Далее каждый ученик должен решить для себя, какую букву передвигать первой (порядок может быть произвольным), какой выбрать маршрут движения (он должен быть кратчайшим).

Партнеры могут передвигать одну и ту же букву, а могут и разные. В этом задании дети также не связаны рамками траектории. Препятствия, расставленные на поле, иногда выглядят как лабиринт. Данный вариант Тяни-Толкай — это логическая задача. Учитель должен перед началом занятия просмотреть все задания, чтобы суметь помочь детям в трудных ситуациях.

Это задание завершает цикл уроков с исполнителем Тяни-Толкай.

БЮРО

ИНТЕЛЛЕКТУАЛЬНЫХ

ТЕХНОЛОГИЙ

БИТ

sound
vision

СРЕДСТВА MULTIMEDIA

настоящее и будущее компьютерного обучения

Sound Vision 8™ сделает доступным для Вас звуковое сопровождение обучающих программ и компьютерные лингафонные курсы

Недорогая звуковая плата полностью совместимая со стандартами **Sound Blaster** и **Adlib Sound**. Запись и воспроизведение звука на частоте до 44 КГц, MIDI-интерфейс. В комплект входят наушники или мини-стерео колонки. С платой поставляется дискета с программами.

Sound Vision™ MPC Upgrade Kit стерео-звук, возможность использовать энциклопедии, справочники, атласы и карты выпускаемые на компакт дисках

Накопитель CD-ROM, 16-битная звуковая стерео-плата с интерфейсом накопителя CD-ROM и MIDI-интерфейсом, микрофон и наушники, активные стерео-колонки, разнообразное программное обеспечение и компакт диски (CD).

АКЦЕССУАРЫ

наушники, микрофоны, накопители CD-ROM, контроллеры CD-ROM, компакт диски (CD).

Е. Б. Ломко

РАЗВИВАЮЩИЕ ИГРЫ ФИРМЫ «НИКИТА»

О компьютере как о средстве «передачи интеллекта» говорят с момента появления самого компьютера. Однако в последнее время в связи с качественным скачком компьютерной и программной технологии этот интерес резко возрос. Развитие систем multimedia, о которых пишут везде и всюду, связывают с созданием обучающих сред.

К сожалению, пока еще смоделировать электронного учителя даже на 486 компьютерах невозможно. Можно сделать электронную иллюстрированную энциклопедию, но нельзя заставить компьютер отвечать на любые вопросы и задавать их, хотя бы потому, что даже мощные алгоритмы сжатия речи «съедают» огромное дисковое пространство, а достойный алгоритм распознавания и моделирования членораздельной речи нам пока не известен.

Несмотря на тотальное распространение в мире компакт-дисков информационной емкостью до 600 МБайт, серийные накопители для них только читают информацию, да и те в России пока диковинка.

Полноценную графику и анимацию (я имею в виду хотя бы 640*480 точек в 256 цветов) умеет программировать далеко не каждый, хотя SVGA-адаптеры уже вытеснили привычные VGA. А без использования расширенной памяти в защищенном режиме работы процессора такая графика в стандартные 640 Кбайт просто не уместится. Какой же эффект от обучения, если на экране ничего не движется и все зеленого цвета?

Поэтому многомиллионная армия учителей может спать спокойно, технократические конкуренты, похоже, не смогут в обозримом будущем заменить живого наставника. На этой оптимистической ноте можно было и закончить, но...

Отрицать роль компьютеров в образовании не менее абсурдно, чем ее преувеличивать. Что же можно и нужно использовать в учебном процессе?

Остановимся подробнее на этой проблеме на примере продукции российской компании «Никита», которая производит **развивающие IBM-совместимые компьютерные игры для детей разного возраста.**

Попробуем расшифровать эти понятия.

1. Почему игры?

Потому что в игровой форме человек косвенно получает значительно больше информации, если, конечно, игра дает такую возможность. Из некомпьютерных примеров наиболее убедительны телевизионные программы «Поле чудес» и «Что? Где? Когда?».

2. Что развивают?

Во-первых, интерес к знаниям. Далеко не

каждый станет музыкантом, художником, ученым, но минимальная польза при постижении общечеловеческих ценностей гарантирована. Игры охватывают такие предметные области, как математика, музыка, рисование, логика, родной и иностранный языки, графическое конструирование, тренировка памяти.

Во-вторых, способность мыслить самостоятельно. Известно, что подавляющее большинство из нас выросло и выучилось по методике «делай, как я». В мире жестокой капиталистической конкуренции детей с младенчества приучают к принятию собственных решений. В итоге раскрепощенные «западные» дети зачастую уступают российским сверстникам в академических познаниях. Я считаю объединение указанных стратегий в системе образования прогрессивным, и именно на этих принципах построены игры «Никиты».

В-третьих, чувство прекрасного. Мир, созданный сценаристами, программистами, мультипликаторами и музыкантами «Никиты» настолько уютен и далек от насилия, что участие в нем скорее приведет к творческому развитию личности, чем к желанию самовыражаться надписями на заборах.

3. Что значит IBM-совместимые?

Все игры работают на стандартных компьютерах следующей конфигурации: процессор 80286, дисплей VGA, мышь. Поддерживается и сетевой вариант «пилотных» классов. Программы могут запускаться как с жесткого диска (если он есть), так и с дискета. Понятно, что на более современных моделях компьютеров игры работают с таким же успехом. Все игры выполнены в прекрасной 256-цветной графике и поддерживают звуковые расширения Sound Blaster и Covox с выносными акустическими системами. Если последних нет, то звук выводится через встроенный громкоговоритель, если тактовая частота компьютера 12 МГц и более, если менее — звук автоматически отключается.

Версий для других компьютеров «Никита» пока не делает, поскольку УКНЦ, КУВТ и «Электроника» имеют тенденцию заменяться на IBM, а прекрасная техника фирмы Apple Computer, к сожалению, еще недостаточно распространена в России.

4. Так для какого же возраста?

Наконец, я подошел к краткому описанию игр «Никиты».

«Азбука-раскраска». Возраст — 3—8 лет.

На самом деле, были случаи, когда и 2,5-летние дети с восторгом играли в эту игру,

но это, скорее, исключение из правила. Программа способствует развитию художественных навыков при изучении алфавита. Главный герой — веселая пчелка — летает по экрану от буквы к букве и раскрашивает кисточкой предметы, которые на эту букву начинаются, предварительно выдавливая краску из тюбиков. Наибольший восторг у детей вызывает возможность перемешивать краски в специальной ванночке.

Серия игр «*Малыш*», как и «Азбука-раскраска», ориентирована на первое знакомство ребенка с компьютером. В этих играх используются методики Марии Монтессори для составления картинок из частей, поиска пар однотипных предметов, первых шагов в арифметике. Наиболее увлекательные и веселые темы — «*Достань банан*» и «*Фоторобот*». В первом случае малыш руководит марширующими мартишками при решении классической задачи «Ханойская башня», а во втором может нарисовать самого себя, соседа или, скажем, учителя, используя набор атрибутов лица: ушей, глаз, носов, ртов и др. В «*Малыше-1*» реализован простейший графический редактор, а в «*Малыше-3*» — серия упражнений для развития зрительной памяти. Возраст — 3—8 лет.

В игре «*Музыкальная шкатулка*» реализован музыкальный редактор с элементами слухового анализа. Это значит, что ребенок может не только поиграть на различных музыкальных инструментах и получить представление о нотной грамоте, но и ответить на вопросы компьютера, связанные с повторением несложных мелодий и угадыванием инструментов. Возраст — 4—9 лет.

«*Кроссворды*» — игра для тех, кто уже пишет, но еще с ошибками. Случайно выбранные слова вставляются в случайно сгенерированные компьютером кроссворды. Слова подбираются в соответствии с пиктограммами, на которых изображены предметы, соответствующие наиболее употребляемым существительным и числительным. Предлагается как русский, так и английский вариант этой игры. Возраст — 5—12 лет.

«*Тренировка памяти*» — серия тестов на запоминание предметов, разноцветных квадратов и косточек домино. Одна из тем — «*Найди лишнее*» — превращается в увлекательнейшее соревнование, и взрослые зачастую уступают в нем детям. Возраст — 5—12 лет.

«*Изучаем часы*» — изучение стрелочных и электронных часов. В программе используется сортировка часов, поиск часов с указанным временем, поиск пары часов с одинаково-

вым временем и др. Наличие нескольких уровней сложности позволяет играть как самым маленьким, так и очень взрослым. Основная направленность игры — развитие умения быстро обрабатывать большой объем разнородной информации.

«*15x15*» — привычные с детства пятнашки предстанут перед вами в несколько иной форме. Здесь необходимо собирать истории в картинках, причём развитие истории происходит по мере составления очередного комикса. Для самых опытных предлагается составление геометрических парадоксов. Получается — «собери несобираемое», т. е. фигура, которую можно увидеть на плоскости, в объеме просто не существует.

«*Наш паровоз*» — 46 увлекательных логических головоломок с железнодорожными составами. Необходимо собирать состав в цветовой последовательности, которая указана в задании. Для этого можно переводить стрелки, подцеплять и толкать вагоны паровозом. Играя, ребенок как бы совершает кругосветное путешествие, постоянно меняется не только пейзаж, но и сложность заданий. Количество уровней позволяет рекомендовать ее как восьмилетним детям, так и восьмидесятилетним старцам.

Еще хотелось бы подчеркнуть простоту интерфейса при освоении этих игр. Вместо привычных меню объектно-графические входы в темы делают процесс познания увлекательным и незаметным. В большинстве игр работает фактор времени и количества, за что, собственно, и начисляются очки, которые впоследствии записываются в таблицу результатов.

Все игры «*Никиты*» прекрасно упакованы и содержат описания и методические рекомендации по использованию в школе и дома.

В заключение хотелось бы отметить, что игры «*Никиты*» обкатаны не только в зарубежных школах и детских садах (право на распространение игр «*Никиты*» приобретено фирмами из Швеции, Израиля, Франции, Швейцарии, Польши), но и во многих школах и детских садах России. В некоторых из них, как, например, московских школе-комплексе «*Царицыно*» и гимназии №1504, преподаватели самостоятельно проработали методики по включению игр «*Никиты*» в учебный процесс.

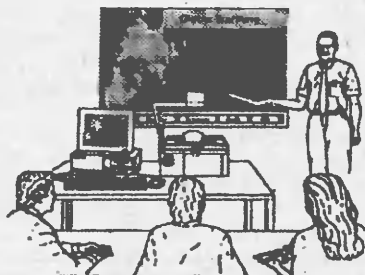
Вообще говоря, не так уж плох компьютер как инструмент «передачи интеллекта».

Контактные

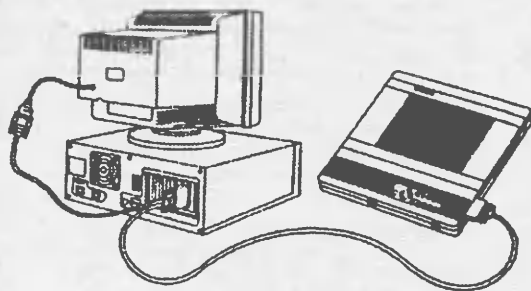
телефоны: (095) 121-42-06
(095) 323-07-37 (факс)

PROXIMA®

Ovation™

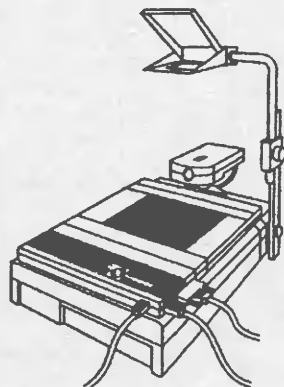


Proxima® Ovation™ – это цветная жидкокристаллическая панель, позволяющая проецировать на большой настенный экран как компьютерное, так и видео изображение, воспроизводя без задержки до 2 миллионов цветов. Установив на проектор панель Ovation, Вы сможете показать видеофильм, компьютерную анимацию или телепередачу самой обширной аудитории. Жидкокристаллическая панель Ovation может быть соединена с такими известными компьютерами как IBM PC, PS/2 и Apple Macintosh, а также с телевизором, видеомagneтофоном, видеокамерой и проигрывателем лазерных дисков, поддерживающими стандарты NTSC, PAL и SECAM. Кроме того, к панели Ovation можно подключить акустические колонки, чтобы обеспечить звуковую поддержку презентации.



Качеством изображения и звука Вы можете управлять с помощью дистанционного пульта управления. Вес жидкокристаллической панели Ovation всего 2.4 кг, что делает ее достаточно удобной при транспортировке.

CYCLOPS™



Proxima® Cyclops™ – это интерактивная система управления презентациями. Система Cyclops состоит из сенсорного глаза и складывающейся указки. Cyclops позволяет человеку управлять компьютерной программой прямо с настенного экрана. Для аудитории система Cyclops выглядит как обычная указка, назначение которой – заменить беспроводную мышь. С помощью указки можно управлять любой программой, использующей мышь. Когда Вы касаетесь экрана и нажимаете кнопку – система Cyclops эмулирует нажатие кнопки мыши в данной части экрана. В дополнение к уже имеющейся указке, Вы можете заказать лазерную указку, работающую на расстоянии до 8 метров. Система Cyclops открывает новые горизонты интерактивных компьютерных презентаций, отличных от стандартных слайд-шоу. Так как система Cyclops выглядит как обычная указка, то ее использование не создает дополнительных проблем даже для пользователей, далеких от техники.

Технические данные

Совместимость с компьютерами

- IBM и совместимые с ними графические адаптеры VGA, S-VGA, VESA, EGA, CGA, MDA
- Семейства Apple Macintosh и PowerBook

Видео совместимость

- NTSC, PAL, SECAM, S-VHS

Соединения компьютерные

- Параллельный DB-25
- Последовательный порт с разъемом Mini DIN 8

Соединения видео

- RCA Composite
- 4-pin Mini DIN S-Video (Y/C) Input Jack

Соединения аудио

- RCA Input Jack для аудио входа
- RCA Input Jack для аудио выхода

Тип панели

- Цветная активная матрица TFT (Thin Film Transistor)

Размер экрана Ovation

- 211 x 158 мм / 171 x 130 мм

Число цветов

- 2 миллиона максимально

Контрастность

- 60 : 1 минимально
- 200 : 1 обычно

Время задержки

- от 30 до 50 миллисекунд

Питающие напряжения

- Автоматическое переключение от 90 до 270 В 47 - 440 Гц

Температура хранения

- от -20° до 60° C

Влажность

- от 10% до 85% без конденсации

Размеры

- Ovation 336мм * 343мм * 35мм
- Cyclops 32мм * 91мм * 57мм

Масса

- Ovation 2.4 кг
- Cyclops 0.4 кг

Комплект стандартной поставки

Проекционная панель Ovation
 Сенсор Cyclops
 Настольный проектор
 Указка - эмулятор мыши
 Лазерный указатель
 Пульт дистанционного управления
 Кабели для соединения с компьютерами IBM и Apple Macintosh
 Сумка для переноски

Достоинства и возможности

Проецирует до 2 миллионов цветов

Передаёт компьютерное и видео изображение с максимальным качеством цветовоспроизведения

Жидкокристаллическая активная матрица

Минимальное время задержки, телевизионное изображение в реальном времени

Совместим с компьютерами, лазерными дисками, телевизорами, видеокамерами

Предоставляет пользователю возможность создавать презентации, включающие компьютерную анимацию, аудио и видео информацию

Обратная проекция

Пользователь может расположить проекционное оборудование за экраном

Поставка кабелей - разветвителей

Синхронность изображения на компьютерном и настенном экране
 Одновременный контроль мышью и специальной указкой

Встроенный вентилятор

Сохранение качества изображения в течение длительного времени

Гарантия - 1 год

За дополнительной информацией и по вопросам поставки обращайтесь



Advanced Instructional
 Software Trading AB

**СРЕДСТВА
ПРЕЗЕНТАЦИИ**
**ЦЕНЫ
В ДОЛЛАРАХ США**
ЖИДКОКРИСТАЛЛИЧЕСКИЕ ПАНЕЛИ

• PROXIMA OVATION 822	10,087.00
• PROXIMA OVATION 814	9,395.85
• PROXIMA OVATION 826	9,352.00
• PROXIMA OVATION 840	7,400.00

ИНТЕРАКТИВНАЯ СИСТЕМА

• SYSCOPS 2030	1,260.00
----------------	----------

ПРОЕКТОРЫ

• MEDIUM 1000	2,175.79
• MEDIUM 5000	1,362.28
• MEDIUM 3600	1,129.80

ЭКРАНЫ

• TRIPOD SCREEN PROFESSIONAL 150x150	275.34
• TRIPOD SCREEN PROFESSIONAL 180x180	384.13

ДОПОЛНИТЕЛЬНЫЕ АКСЕССУАРЫ

• ВИДЕОАДАПТЕР A194	1,048.53
• ЛАЗЕРНАЯ УКАЗКА	477.00
• НАБОР ДЛЯ ЧИСТКИ	28.88
• ЛАМПЫ	
LAMP 36V/400W	25.00
LAMP 575 HMI	448.19
• СУМКИ ДЛЯ ПЕРЕНОСКИ	
A278S	256.79
A289H	404.28

ПРОГРАММНЫЕ СРЕДСТВА

• MULTIVISION 4/2	199.00
-------------------	--------

Все цены включают НДС; оплата в рублях по текущему курсу ЦБ РФ

 За дополнительной
информацией и
по вопросам поставки
обращайтесь

**Advanced Instructional
Software Trading AB**

PRICE LIST

В. А. Урнов,

генеральный директор AIST AB

Е. С. Крылова,

«Проект «Образование» НЦПСО

МЫ ПОМОЖЕМ ВАМ УСПЕТЬ В XXI ВЕК

Едва ли не самая большая трудность на нынешнем этапе внедрения современных информационных технологий в различные области человеческой деятельности, в частности в образование, — недостаточно четкое понимание термина, «мультимедиа». Чаще всего под мультимедиа подразумевают попытку использовать всевозможные виды информации одновременно: звуковую, телевизионную, компьютерную, графическую, анимационную, текстовую — из всех возможных источников.

Крупнейшее мировое событие 1993 г. в области информационных технологий — выставка COMDEX в Лас-Вегасе, где мы имели честь присутствовать, — отчетливо показала, куда устремлен вектор их развития, а именно: все крупнейшие производители обозначили недвусмысленный поворот в направлении мультимедиа. Это такие ведущие корпорации, как INTEL, MICROSOFT, BORLAND, DELL, GATEWAY, CREATIVE LAB и многие другие фирмы.

Сегодня уже ни DELL, ни GATEWAY — а это сейчас крупнейшие производители компьютеров в мире — практически не выпускают компьютеры без CD ROM, который становится составной частью компьютера. INTEL и CREATIVE LAB, самые крупные производители всего, что связано с компьютерным видеоборудованием, совмещением теле/видео и компьютерного изображения, приложили огромные усилия и сделали действительно доступный по цене и простоте использования видеобластер — устройство, которое позволяет оцифровывать видеoinформацию, перемешивать и затем использовать ее в компьютере, сбрасывать обратно на

видео. Другими словами, обеспечивает обмен «видео — компьютер».

На то же самое нацелена и вся перспективная идеология развития самого компьютера. Не случайно в мире растет потребность во все более быстродействующих процессорах, более мощных машинах. Ведь обработка видеoinформации требует совсем другого уровня скоростей, других объемов памяти. Другого уровня мышления, наконец.

Как же должна ответить на вызов времени система образования? В Научном центре программных средств обучения (НЦПСО) готов ответ на этот вопрос. Наше решение ориентировано на активное и многоплановое использование программно-аппаратного комплекса MultiVision PRO.

Проработав на стезе школьной информатики довольно долго, первые два года мы пытались, как все, двигаться прямым, экстенсивным путем. Писали обучающие программы, тренажеры.

На этом пути нами был наработан достаточно большой опыт, и в первую очередь опыт неудач, причем идеологического порядка, а не в реализации этих идей. Поэтому около трех лет назад в НЦПСО поставили задачу разработать другие подходы, концепции использования информационных технологий в обучении. В результате этих поисков и появился MultiVision PRO.

MultiVision PRO — это комплекс взаимосвязанных программных и аппаратных решений, каждое из которых воплощает и поддерживает самые перспективные, новые идеи и технические решения на мировом уровне.

Основой комплекса является компьютер

IBM PC/486. Уникальные демонстрационные возможности комплекса базируются на использовании разработок, выполненных в 1992 г. калифорнийской фирмой PROXIMA, — цветной жидкокристаллической панели OVATON, позволяющей проецировать на большой настенный экран как компьютерное, так и видеозображение, воспроизводя без задержки до 2 миллионов цветов, и интерактивной системы управления демонстрациями CYCLOPS, состоящей из сенсорного «глаза» и лазерных указок двух видов, позволяющих управлять компьютером издали: с помощью одной — стоя у экрана, лицом к аудитории, а другой — находясь от экрана на расстоянии нескольких метров.

Мощный поток света излучает проектор западногерманской фирмы MEDIUM — проекторов такого уровня несколько лет назад просто не существовало.

Программная среда MultiVision, которая одушевляет комплекс, создана в русле самых современных идей программирования. Ее модификация на русском языке — это не перевод, а именно адаптированный вариант и программы, и документации.

Система школьного образования во всем мире находится в начале революционных изменений, связанных с воздействием современных информационных технологий мультимедиа. Информационные технологии и информатика как школьный предмет отнюдь не тождественны, а внедрение информационных технологий не тождественно установке компьютерных классов в один-два кабинета, даже если это очень хорошая техника. На наш взгляд, главная задача состоит в том, чтобы сделать современные информационные технологии реальным инструментом обучения в любой предметной области.

В своей концепции использования MultiVision PRO мы исходим из того, что создание компьютерных учебных курсов под конкретные методики бесперспективно. Ведь что бы ни говорилось о переносимых методах и стандартах обучения, реальный образовательный процесс очень сильно привязан к

личности учителя и другим фактическим составляющим. И никаких производственных и финансовых ресурсов не хватит, чтобы подготовить курсы достаточной степени гибкости и необходимого объема.

В традиционном процессе обучения используются учебники, которые содержат не ориентированную на персоналии выжимку необходимого, стандартизованного набора знаний; они служат лишь базой, отправной точкой для изучения материала.

Если же не пытаться персонализировать информационные технологии, то смысл их внедрения просто теряется, так как теряются гибкость, адаптивность, которые и составляют их преимущество.

Переход к современным информационным технологиям мультимедиа от печатных изданий, возможно, даже более принципиален, чем переход от рукописей к печатному слову.

Объем сохраняемой и передаваемой информации, скорость передачи информации и ее обработки, способы хранения информации и доступа к ней, возможность обратной связи, прямого диалога с источником информации — это качественно новые параметры при работе с информацией, в нашем случае, — с учебной.

В традиционном учебном процессе учитель (но не учебник) организует и поддерживает обратную связь с учеником. Именно обратная связь и определяет необходимость персонализации современных информационных технологий. Отсюда — та самая невозможность создавать компьютерные учебные курсы под конкретные методики.

Комплекс Multi-Vision PRO включает инструментальную оболочку MultiVision, которая предоставляет учителю, (либо самому, либо с помощью учеников) богатые возможности для гибкой генерации учебных курсов. Он технически позволяет быстро апробировать, корректировать эти курсы и сразу же использовать их в процессе обучения. Таким образом, комплекс дает возможность как создания учебных программ, так и их незамедлительного использования.

Подготовка и использование программного

MultiVision PRO - это комплекс взаимосвязанных программных и аппаратных решений, каждое из которых воплощает и поддерживает самые перспективные, новые идеи и технические решения на мировом уровне.

продукта при помощи комплекса характеризуется достаточно большой легкостью, простотой, гибкостью, скоростью выполнения подготовительной работы и ее высоким качеством, недоступным непрофессионалам в других системах программирования, а также универсальностью, можно сказать, предметной независимостью.

В случае необходимости в программу можно внести коррективы. Например, в параллельных классах уроки могут выглядеть по-разному. Причем большая гибкость в подготовке программного продукта делает возможным использование полученных курсов другими учителями.

Возможно, тот, кто не видел MultiVision PRO в работе и не знаком с ним по техническим описаниям, отнесется к написанному выше с определенной долей скептицизма, подумает, что это звучит слишком хорошо, чтобы быть правдой. Однако же комплекс существует.

Главная идея применения информационных технологий на базе комплекса — вернуть учителю центральную роль в процессе обучения.

Учитель утратил ведущие позиции, когда каждый ученик повернулся к нему спиной, сел лицом к экрану и занялся своей программой. Если бы это еще давало результат! Однако, опыт показал, что, как только учитель потерял реальную возможность управлять уроком, эффективность обучения резко снизилась. Учитель «выпал» из урока, и урока не стало. Комплекс MultiVision PRO позволяет учителю занять свое привычное место — у доски-экрана, с указкой-инструментом в руке, лицом к аудитории, управляя ее вниманием и восприятием. Современные информационные технологии становятся достоянием не узкого круга специалистов в области информатики, а каждого учителя-предметника.

В каждом предмете есть темы, которые целесообразно преподавать, используя мультимедиа. Ведь в ряде случаев, чтобы усвоить то или иное понятие, получить представление о структуре того или иного явления, динамике того или иного процесса, их просто необходимо увидеть. Или услышать. Причем, например, в исто-

рии, географии, биологии, иностранном языке таких тем достаточно много, а в других, таких, как математика, их гораздо меньше. Это разделение довольно естественно и соответствует уровню абстракции самой науки, роли наглядности и фактографии в ее постижении.

Там, где роль наглядности велика — исторические реалии, флора и фауна климатических зон, карты, процессы жизнедеятельности организма, представленные в динамике, химические реакции, сопровождающиеся видимыми изменениями, произношение звука, интонация во фразе, анализ музыкальных стилей и так далее, — вопрос о применении информационных технологий мультимедиа нами принципиально решен, понятны пути его реализации, и здесь MultiVision PRO трудно переоценить. В этих предметах и темах мы

идем от демонстрации, разъяснения, от наглядности, от научения упорядочиванию материала, созданию собственных концепций ученика на знаниях, которые он получает.

Наверное, кто-то спросит зачем, собственно, огород городить и тратить колоссальные средства? Ведь хороший учитель и традиционными средствами отлично обходится!

Да, конечно, хороший учитель может без

многого обойтись, может быть, даже без доски и мела объяснит все «на пальцах». Но вряд ли кто-то всерьез будет спорить, что, имея доску и мел, он все-таки научит лучше, с большей отдачей, эффективнее. Больше успеет, доходчиве изложит.

При проектировании комплекса стояла задача: создать инструмент обучения, который позволил бы учителю работать гораздо эффективнее, чем в традиционной схеме. Настоятельно лучше, чтобы это оправдало высокую стоимость комплекса. Чтобы учитель с удовлетворением ощутил дополнительные степени свободы, совершенно новые возможности.

Сегодня на основании опыта работы учителей с комплексом, многочисленных демонстраций и докладов на выставках и конференциях, встреч и обсуждений с педагогами и

*Комплекс MultiVision PRO
включает инструментальную
оболочку MultiVision, которая
предоставляет учителю
(либо самому, либо с помощью
учеников) богатые
возможности для гибкой
генерации
учебных курсов.*

специалистами в области информатики в Москве и других городах у нас появилась надежда и даже, пожалуй, уверенность в том, что наш комплекс — вещь заведомо не менее полезная, чем доска и мел.

Какой, собственно, инструмент получает пользователь MultiVision PRO? Он получает в руки устройство, позволяющее быстро, красиво, профессионально и эффективно писать интерактивные учебные курсы, получает возможность при помощи уникальных демонстрационных устройств показывать и использовать свое собственное программное обеспечение чуть ли не в момент его изготовления, причем не на мониторе компьютера, а на большом экране. К тому же учитель может управлять работой своей программы традиционной указкой. Не менее важна возможность быстро и гибко менять все, что он сделал, и что еще очень важно — результат полностью отчуждаем, то есть он может быть безболезненно перенесен на любой такой же компьютер простейшим способом — на дискете и использован в любой такой же демонстрационной системе.

В НИЦПСО образовано специальное подразделение «Проект «Образование» для поддержки учителя, работающего с комплексом MultiVision PRO. Эту поддержку нужно оказывать с двух сторон: путем обучения работе с системой и поддержки разработками-полуфабрикатами, чтобы максимально разгрузить учителя при подготовке материала. Речь идет о картинках, полноэкранных изображениях, мультфильмах, текстах, музыкальных фрагментах и многом другом, в том числе об определенных технологических решениях. Этот материал должен быть у учителя под рукой, чтобы он мог на основе полуфабрикатов быстро сгенерировать тот или иной учебный курс.

Несколько слов об информатике как учебном предмете. Мы считаем, что применение MultiVision PRO наполняет самое информатику абсолютно другим содержанием. У человека не должно возникать дискомфорта или тем более комплексов при работе с устройствами, вошедшими в обиход современной цивилизации.

MultiVision — это язык, но язык гораздо

более высокого уровня, чем любой язык программирования, будь то BASIC, PASCAL или C++. Причем, изучая данный язык, ученик встречается со всеми базовыми принципами обработки информации.

В среде MultiVision существует практически весь набор компьютерных средств, с которыми надо научить работать пользователя. Здесь есть очень хороший графический редактор, аниматор профессионального уровня, текстовый процессор и другие инструментальные модули. Причем работа с этими модулями дает результат практически профессионального качества, а, в отличие от других пакетов, они просты в обращении. Следовательно, давая эту систему в руки ученику как базовую среду, мы решаем сразу множество задач.

Во-первых, мы готовим его как пользователя.

Во-вторых, — обучаем его базовым принципам обработки информации с точки зрения языкового, структурного, но не узкопрограммистского подхода.

В-третьих, — самое главное — мы формируем у ученика созидательный мотив. Более того, этот мотив усиливается возможностью немедленной реализации, а также использо-

ванием результата для работы на уроках по другим предметам, на которых используется комплекс MultiVision PRO. ■

В-четвертых, эта система позволяет сосредоточить основное внимание непосредственно на предметной области.

До сих пор при попытке поручить ученикам писать программы в предметной области на BASIC или, скажем, на PASCAL главным для ученика оставалось найти программистское решение поставленной перед ним задачи.

При работе с MultiVision PRO ученик может на это не отвлекаться, а сосредоточиться на предметной области и существе вопроса. Это инициирует и более глубокое изучение указанной предметной области.

В-пятых, работая в среде MultiVision

*В НИЦПСО
образовано
специальное подразделение
«Проект «Образование»
для поддержки
учителя, работающего
с комплексом
MultiVision PRO.*

PRO ученик получает возможность создать профессионально сделанную программу, а профессионально сделанная вещь — это очень сильная мотивация.

Грамотное использование MultiVision PRO позволяет на уроках информатики готовить квалифицированных пользователей, понимающих суть работы с информацией, владеющих идеологией и практикой современных информационных технологий мультимедиа.

Таким образом, MultiVision PRO поддерживает все существующие сегодня концепции преподавания информатики в школе.

В 1994/95 учебном году мы сделаем дальнейший шаг, суть которого коротко заключается в следующем.

Все наши партнеры по этой разработке движутся вперед. Мы полагаем, что сможем поставлять в составе комплекса PENTIUM-компьютер.

Мы готовы использовать технологию CD ROM и будем поставлять компьютеры со встроенным устройством управления (CD ROM drive).

Это могут быть географические атласы, атласы по биологии, некоторые экономические, исторические, изобразительные материалы. Из них мы делаем подборки по 5—10 дисков и включим их в

состав комплекса MultiVision PRO — ведь компьютеры, входящие в состав комплекса, будут иметь встроенные CD ROM drive. Эти материалы мы рассматриваем как дополнительный источник черновой информации для учителя, полуфабрикатов для создания им собственных учебных курсов.

Мы обязательно оставим в системе цветное сканирующее устройство, которое позволяет сразу обрабатывать информацию, получать заготовку для создания курса с листа, будь то учебник или какой-то другой предмет. И мы уже начали поставлять в составе комплекса более простую в работе, нежели 2С, модель 2СХ.

В качестве лазерных принтеров мы начинаем поставлять HP 4L. Сегодня это самая несложная в использовании модель. Решен вопрос с русской шрифтовой поддержкой.

Кроме того, огромный шаг вперед сде-

лала фирма PROXIMA, которая приступила с января 1994 г. к серийному производству устройства DESKTOP PROJECTOR, без преувеличения, самого последнего достижения в мире в области проекционной техники. DESKTOP PROJECTOR объединяет в себе одновременно проектор, OVATION на 2 млн. цветов, CYCLOPS, звуковые колонки. Модель гораздо более приемлема для использования в классе и гораздо более высокого качества за счет того, что проектор и OVATION помещены в один корпус и в этой конструкции до микронов рассчитаны все светопотоки.

Используя самый мощный из существующих проекторов и OVATION, мы можем получить на большом экране порядка 50—60 люменов. Используя школьный комплекс, на экра-

не можно получить около 25-30 люменов. Даже это изображение вполне годится для работы без полного затемнения в классе. Единственное условие — на экран не должны падать прямые лучи света.

DESKTOP PROJECTOR дает минимум 110 люменов на экране при непрямом освещении. При этом CYCLOPS, заранее установленный на DESKTOP PROJECTOR, сразу же снимает все проблемы с настройкой, так как она

происходит один раз и только по реперным углам, после этого сам DESKTOP PROJECTOR можно поворачивать, поднимать, опускать — все параметры настройки CYCLOPS сохраняются, потому что он жестко закреплен и будет продолжать работать.

Уязвимость — это хотелось бы подчеркнуть — уязвимость системы станет на порядок ниже. Если раньше лежащий на проекторе OVATION можно было случайно сбить локтем и потерять на этом \$10 000, потому что OVATION, естественно, расколется, то уронить DESKTOP PROJECTOR — примерно так же трудно, как уронить телевизор.

Встроенный внутрь DESKTOP PROJECTOR источник звука делает его более удобным в работе, а также позволит

*В среде MultiVision
существует
практически весь набор
компьютерных средств,
с которыми надо
научить работать
пользователя.*

сохранить динамики. Колонки, которые мы поставляем, очень маленькие. Дети любят забирать их в качестве сувениров. При использовании DESKTOP PROJECTOR колонки можно унести только со всей системой. Расположены колонки удачно, так, что при нормальной установке DESKTOP PROJECTOR они смотрят навстречу аудитории.

Мы оставим в комплекте и будем обязательно поставлять и впредь видеосистему, точнее, видеоманитфон. Он напрямую подключается к DESKTOP PROJECTOR. При этом необходимость телевизора в составе комплекса, вообще говоря, отпадает. Качество DESKTOP PROJECTOR при проектировании изображения на стену не уступает телевизионному, а во многих отношениях просто полезнее его, потому что это отраженный, а не излученный свет. Правда, чтобы эта техника работала как телевизор, причем качественный телевизор с диагональю от 92 см до 3,6 м нужно будет поставлять видеоманитфон с тюнером.

В конце 1994 г. будет закончен пакет, который позволит при помощи установленного дополнительного аппаратного оборудования, например, INTEL SMART

VIDEORECORDER, пересылать информацию из видеоманитфона в компьютер и обрабатывать ее так же, как сейчас обрабатывается графическое компьютерное изображение. Или почти так же. Мы это делаем и, безусловно, сделаем.

Эти дополнительные видеовозможности позволяют использовать в качестве фона демонстрации не просто статическое или динамическое компьютерное изображение, а, например, фильм.

Со вторым вариантом комплекса, т. е., с DESKTOP PROJECTOR, с дополнительными звуковыми возможностями и с CD ROM мы, видимо, будем готовы выйти на рынок к середине 1994 г.

Следующую, третью модификацию

MultiVision PRO с дополнительными видеовозможностями мы планируем подготовить к началу 1995 г.

Если кто-нибудь скажет, что использование IBM-совместимого компьютера для нужд образования в таком решении продолжает проигрывать использованию компьютера APPLE MACINTOSH, то начиная со второго этапа развития MultiVision PRO мы готовы с ним поспорить.

Более того, мы готовы отстаивать превосходство нашего решения и будем правы уже потому, что попытка использовать здесь компьютер другого типа, допустим MACINTOSH, повлечет за собой минимум 2—3-летнюю работу по созданию для него российско-ориентированной, русскоязычной программной, со-

держательной и другой необходимой информационной поддержки. В частности, потребуются издание массы различных книжек и другой литературы.

Подводя краткий итог всему написанному выше, хотелось бы еще раз подчеркнуть, что у нас есть, чем ответить на вызов времени, мы предлагаем на бурно развивающемся рынке информационных технологий свои апробированные решения. У нас есть специали-

сты, всесторонне поддерживающие комплекс, есть идеи дальнейшего совершенствования этих технологий каквширь, так и вглубь.

Мы выбираем мультимедиа, верим в глубокий смысл того, что делаем, намерены и дальше поставлять наши технологии в школу.

Давайте вторгнемся в XXI век вместе!

Телефоны для справок:

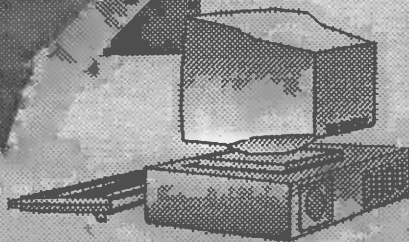
(095) 272-76-71 (095) 214-46-49

(095) 214-77-84 (095) 214-54-33

*Мы выбираем
мультимедиа,
верим в глубокий смысл того,
что делаем,
намерены и дальше
поставлять
наши технологии
в школу.*

*Это самый дорогой,
но успешный способ доставки!*

MultiVision Pro



ВЫ ХОТИТЕ:

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

ТОГДА ВАМ НЕОБХОДИМ:

комплекс

MultiVision Pro



Advanced Instructional Software Trading AB

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

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

MultiVision Pro

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



CANADA

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



U. S. A.

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



U. S. A.

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



U. S. A.

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



SWEDEN

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



GERMANY

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



JAPAN

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

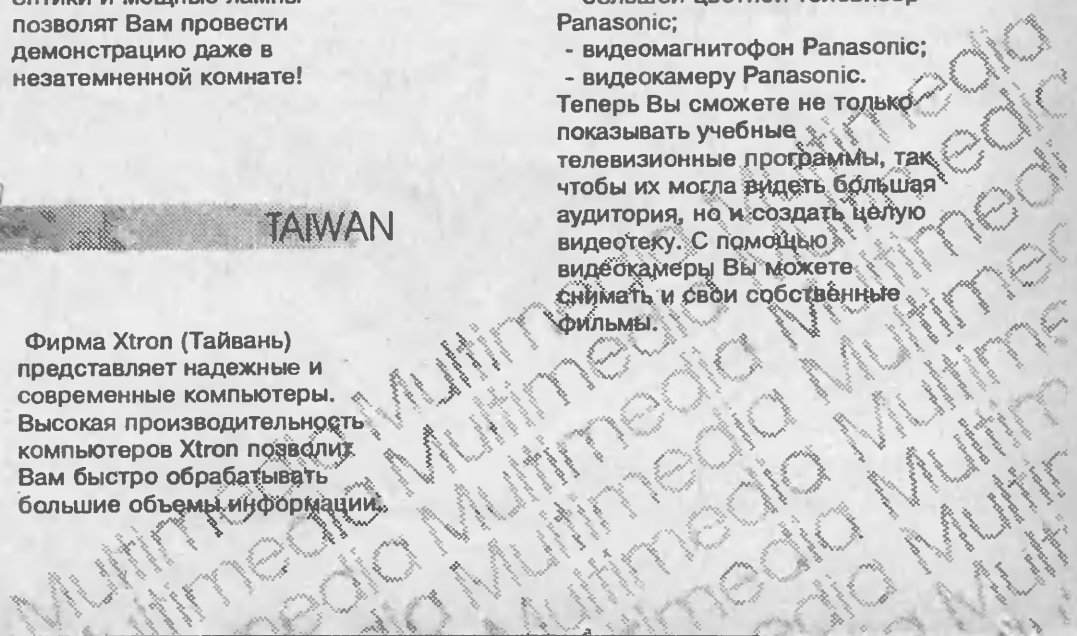
- большой цветной телевизор Panasonic;
- видеомагнитофон Panasonic;
- видеокамеру Panasonic.

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



TAIWAN

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





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

MultiVision Pro

ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР

В. О. Байбаков

ОРГАНИЗАЦИЯ ИНТЕРФЕЙСА ПРОГРАММЫ НА ЯЗЫКЕ QUICK BASIC

Вначале договоримся, что мы будем понимать под словом «интерфейс». Интерфейс (interface) — слово английское. Состоит оно из двух частей: inter (префикс, имеющий значение взаимодействия) и face (лицо). То есть, проще говоря, это все то, что стоит между двумя лицами и обеспечивает их взаимодействие. В качестве «лиц» могут выступать ваша или какая-нибудь другая программа (программные средства), различные внешние и внутренние устройства компьютера (аппаратные средства) и, наконец, человек-оператор. В качестве интерфейса тоже могут быть программные, аппаратные средства и человек-оператор (если в качестве конечного «фейса» рассматривать его начальника).

Львиную долю труда по составлению программы занимает разработка интерфейса. По тому, как он сделан, можно с уверенностью судить о квалификации программиста, а по интерфейсу фирменных программных продуктов — о годе их создания. Так, до 1987-89 гг. для общения с пользователем применялись либо специальный язык команд (SUPERCALC-3, GWBASIC, DBASE-2 и 3 и др.), либо «горячие клавиши», когда команды выполняются при нажатии на определенные клавиши (например, PCTOOLS). Эти методы сохранились и сейчас, но носят чаще всего вспомогательный характер. (Пожалуй, за исключением программы PCTOOLS, сохранившей интерфейс еще с тех времен).

В конце восьмидесятых годов наиболее широкое распространение получило управление по принципу «меню», когда с помощью клавиш управления курсора осуществляется выбор одного из нескольких альтернативных вариантов действий. И наконец, в последнее время стало модным помещать все меню и сообщения в раскрывающиеся окна с «тенью».

Чтобы разобраться в возможностях организации современного интерфейса средствами Quick BASIC, давайте разберем программу, которая проверяет и изменяет состояние ключа Caps Lock. Эта программа интересна тем, что она дает повод поговорить о том, как программным способом определять и изменять состояние ключей клавиатуры и содержит:

- подпрограмму WIND для организации «раскрывающихся» окон с «тенью»;
- подпрограммы PROMPT и MENU для организации меню (если вы знакомы с системами FOXBASE или CLIPPER, вы заметите, что идея организации меню заимствована отсюда);
- маленькую, но очень удобную подпрограмму SAY для вывода сообщений в нужную точку экрана.

'Объявление подпрограмм. Вам это делать не обязательно,

' Quick BASIC вставит эти строки сам

```
DECLARE SUB wind (yn!, xn!, yk!, xk!, cp!, cf!, shand!)
```



```

DECLARE SUB prompt (y!, x!, p$)
DECLARE SUB menu (a!, b!, cpo!, cfo!, cpv!, cfv!)
DECLARE SUB say (y!, x!, a$)

```

' Следующие три строки необходимы для работы
' подпрограмм организации меню

```

DIM promptx(32), prompty(32), promp$(32)
COMMON SHARED number, promptx(), prompty(), promp$()
number = 0

```

' Пишем заставку

```
COLOR 0, 7
```

```
CLS
```

```
CALL say(2, 20, "Учебная программа. (С) В.Байбаков 1992")
```

```
CALL say(23, 10, "Используя клавиши управления курсором, укажите нужный пункт меню")
```

```
CALL say(24, 5, "и нажмите [Enter]. Можно просто нажать клавишу с номером нужного пункта.")
```

' Рисуем окошко, задаем тексты пунктов меню и координаты их начала

```
CALL wind(5, 20, 10, 59, 14, 1, 1)
```

```
CALL prompt(6, 23, "1. Проверить состояние Caps Lock")
```

```
CALL prompt(7, 23, "2. Включить Caps Lock      ")
```

```
CALL prompt(8, 23, "3. Выключить Caps Lock      ")
```

```
CALL prompt(9, 23, "4. Завершить работу        ")
```

' Вызываем подпрограмму MENU , в результате работы которой получаем номер выбранного пункта или ноль, если нажата клавиша [Esc]. Обратите внимание: поскольку второй параметр при вызове подпрограммы отличен от нуля, после окончания ее работы информация о пунктах меню сохраняется.

```
doing: CALL menu(a, 1, 14, 1, 14, 4)
```

```
SELECT CASE a
```

```
    CASE 1 ' Проверка состояния ключа Caps Lock
```

' Читаем информацию о состоянии ключевых клавишей

```
DEF SEG = 0
```

```
c = PEEK(&H417)
```

```
DEF SEG
```

' Анализируем состояние выбранного ключа и выводим сообщение

```
IF c AND 64 THEN d$ = "включен" ELSE d$ = "выключен"
```

```
CALL wind(15, 20, 17, 59, 14, 4, 0)
```

```
CALL say(16, 33, "Caps Lock " + d$)
```

```
CASE 2 ' Включаем режим Caps Lock
```

```
DEF SEG = 0
```

```
POKE &H417, (PEEK(&H417) OR 64)
```

```
DEF SEG
```

```
' Выводим сообщение
```

```
CALL wind(15, 20, 17, 59, 14, 4, 0)
```

```
CALL say(16, 33, "Caps Lock включен")
```

```
CASE 3 'Выключаем режим Caps Lock
```

```
DEF SEG = 0
```

```
POKE &H417, (PEEK(&H417) AND 191)
```

```
DEF SEG
```

```
' Выводим сообщение
```

```
CALL wind(15, 20, 17, 59, 14, 4, 0)
```

```
CALL say(16, 33, "Caps Lock выключен")
```

```
CASE ELSE
```

```
STOP
```

```
END SELECT
```

```
' Ожидаем нажатия любой клавиши
```

```
a$ = INPUT$(1)
```

```
' Стираем сообщение и продолжаем работу
```

```
CALL wind(15, 20, 17, 59, 7, 7, 0)
```

```
GOTO doing
```

```
END
```

```
SUB prompt (y, x, p$)
```

```
' Подпрограмма записывает информацию о пункте меню в специально
```

```
' выделенные массивы (promptx, prompty и promp$) и увеличивает
```

```
' на 1 счетчик заданных пунктов — number
```

```
' x и y — координаты начала текста пункта меню
```

```
' p$ — текст сообщения
```

```
number = number + 1
```

```
promptx(number) = x
```

```
prompty(number) = y
```

```
promp$(number) = p$
```

```
END SUB
```

SUB menu (a, b, cpo, cfo, cpv, cfv)

- ' a - номер выделенного пункта меню, после завершения работы подпрограммы - номер выбранного пункта меню или 0, если нажата клавиша [Esc]
- ' b - если подпрограммой получен 0, после окончания ее работы информация о пунктах меню будет уничтожена
- ' cpo - номер цвета переднего плана невыделенных пунктов меню
- ' cfo - номер цвета фона невыделенных пунктов меню
- ' cpv - номер цвета переднего плана выделенного пункта меню
- ' cfv - номер цвета фона выделенного пункта меню
- ' number - число пунктов меню

- ' Назначаем номер пункта меню, который будет выделен первым.
- ' Можно убрать эту строку и задавать номер первого выделяемого пункта из вызывающей программы.

a = 1

- ' Пишем невыделенные пункты меню

```
COLOR cpo, cfo
FOR i = 1 TO number
  LOCATE prompt(i), promptx(i)
  PRINT promp$(i);
NEXT i
```

- ' Выделяем один из пунктов меню цветом

```
prompt1: LOCATE prompt(a), promptx(a)
COLOR cpv, cfv
PRINT promp$(a);
```

- ' Ожидаем нажатия клавиши

```
prompt2: a$ = INKEY$
IF LEN(a$) = 0 GOTO prompt2
```

- ' Снимаем выделение пункта меню

```
COLOR cpo, cfo
LOCATE prompt(a), promptx(a)
PRINT promp$(a);
```

- ' Анализируем полученный сигнал

```
IF LEN(a$) = 2 THEN ' Нажата одна из клавиш управления
  a$ = RIGHT$(a$, 1) ' курсором (с двойным кодом)
  SELECT CASE ASC(a$)
```

```
CASE 72, 75 ' Нажата стрелка вверх или влево
  a = a - 1
  IF a < 1 THEN a = number
CASE 80, 77 ' Нажата стрелка вниз или вправо
  a = a + 1
  IF a > number THEN a = 1
CASE 71 ' Нажата клавиша [Home]
  a = 1
CASE 79 ' Нажата клавиша [End]
  a = number
CASE ELSE
  END SELECT
ELSE ' Нажата иная клавиша (с одиночным кодом)
  SELECT CASE ASC(a$)
    CASE 13 ' Нажата клавиша [Enter]
      GOTO promptend
    CASE 27 ' Нажата клавиша [Esc]
      a = 0
      GOTO promptend
    CASE ELSE ' Проверяем, не нажата ли клавиша с
      ' первым символом одного из пунктов меню
      FOR i = 1 TO number
        IF ASC(a$) = ASC(prompt$(i)) THEN
          a = i
          GOTO promptend
        END IF
      NEXT i
    END SELECT
  END IF
  GOTO prompt1
promptend:
IF b = 0 THEN number = 0 ' Уничтожаем информацию о пунктах
  ' меню, обнуляя их счетчик
END SUB
```

```
SUB say (y, x, a$)
```

```
' Подпрограмма выводит сообщение в заданную точку экрана
```

```
  LOCATE y, x
  PRINT a$;
END SUB
```

```
SUB wind (yn, xn, yk, xk, cp, cf, shand)
```

```
' Подпрограмма рисует "окно" с рамкой и "тенью"
```

```
' уп, хп - координаты левого верхнего угла окна
' ук, хк - координаты правого нижнего угла окна
' ср, cf - цвет рамки и поля окна
' shand - если в подпрограмму передана 1, рисуется "тень"
```

```
COLOR ср, cf
```

```
' Определяем координаты центра окна
```

```
xs = INT((хк + хп) / 2)
ys = INT((ук + ун) / 2)
```

```
' Рисуем раскрывающееся окно
```

```
FOR x = xs TO хп + .5 STEP -1
yt = (ук - ун) / (хк - хп) * (x - хп) + ун
ytk = 2 * (ys - yt) + yt
FOR y = yt TO ytk
LOCATE y, x
PRINT SPACE$(2 * (xs - x))
NEXT y, x
LOCATE ун, хп
```

```
' Рисуем рамку и, если shand=1 - "тень"
```

```
PRINT "┌"; STRING$(хк - хп - 3, 205); "┐ "
FOR y = ун + 1 TO ук - 1
LOCATE y, хп
PRINT "│"; SPACE$(хк - хп - 3); "│";
IF shand = 1 THEN COLOR 0, 0: PRINT " "
COLOR ср, cf
NEXT y
LOCATE ук, хп
PRINT "└"; STRING$(хк - хп - 3, 205); "┘ ";
IF shand = 1 THEN
COLOR 0, 0
PRINT " "
LOCATE ук + 1, хп + 2
PRINT SPACE$(хк - хп - 1)
COLOR ср, cf
END IF
END SUB
```

В заключение давайте немного поговорим о том, как можно программным способом определять и изменять состояние ключей клавиатуры. Получить информацию о состоянии ключей Insert, Caps Lock, Num Lock и Scroll Lock, а также ключевых клавиш можно, прочитав два байта памяти с адресами &H417 и &H418. Надо только предварительно установить нужный сегмент памяти. Не забудьте потом восстановить его старое значение, иначе могут быть неприятности. Делаем это так:

```
DEF SEG = 0
```

```
c = PEEK(&H417)
```

```
d = PEEK(&H418)
```

```
DEF SEG
```

Теперь разберем, как расшифровать полученную информацию. Не вдаваясь в тонкости двоичной системы счисления, отметим только, что если условие

```
IF c AND a THEN ...
```

истинно, то в зависимости от значения a:

128 - включен режим Insert;

64 - включен режим Caps Lock;

32 - включен режим Num Lock;

16 - включен режим Scroll Lock;

8 - в данный момент нажата клавиша Alt;

4 - в данный момент нажата клавиша Ctrl;

2 - в данный момент нажата левая клавиша Shift;

1 - в данный момент нажата правая клавиша Shift.

Если истинно условие

```
IF d AND a THEN ...
```

то, в зависимости от значения a:

128 - в данный момент нажата клавиша Insert;

64 - в данный момент нажата клавиша Caps Lock;

32 - в данный момент нажата клавиша Num Lock;

16 - в данный момент нажата клавиша Scroll Lock.

Если вам необходимо изменить состояние одной из ключевых клавиш, надо получить значение байта &H417, изменить его и вновь передать в память :

```
DEF SEG = 0
```

```
POKE &H417,(PEEK(&H417) OR a)
```

```
DEF SEG
```

чтобы включить режим, или:

```
DEF SEG = 0
```

```
POKE &H417,(PEEK(&H417) AND f)
```

```
DEF SEG
```

чтобы выключить его. Значения a и f равны:

для Insert - a=128 , f=127;

для Caps Lock - a= 64 , f=191;

для Num Lock - a= 32 , f=223;

для Scroll Lock - a= 16 , f=239.

Отметим одну особенность компьютеров типа XT. У них связь клавиатуры с системным блоком односторонняя, а поэтому, когда вы изменяете состояние ключей Caps Lock, Num Lock или Scroll Lock программным путем, соответствующие этим ключам лампочки на блоке клавиатуры остаются в прежнем состоянии. У компьютеров типа AT эта связь двусторонняя, и лампочки всегда показывают истинное состояние соответствующего ключа, независимо от того, изменялось ли оно программным путем или нажатием на клавишу.



БЮРО

ИНТЕЛЛЕКТУАЛЬНЫХ

ТЕХНОЛОГИЙ

Обучающие и деловые компьютерные системы для IBM

Blank Editor 1.0

новый редактор бланков для школьной канцелярии

- многооконный интерфейс
- система контекстной помощи
- встроенный текстовый редактор с русификатором
- поддержка мыши



- позволяет создавать и редактировать текстовые бланки любой сложности
- запись заполненных бланков в файл или вывод на принтер
- хранение ранее введенной информации в базе данных
- набор готовых бланков для школьной канцелярии и бухгалтерии
- набор форм договоров

Наша программа возьмет на себя рутинную работу и сэкономит Ваше время

Auto Control

конструктор тестов

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

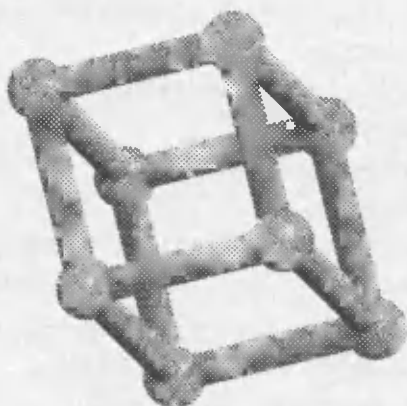
Эти возможности делают Auto Control наилучшим выбором среди контролирующих систем

Constructive Geometry

учебная среда

Пакет включает набор открытых для пользователя модулей:

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



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



535-2222, 534-4831, факс 534-4832

БИТ

БЮРО
ИНТЕЛЛЕКТУАЛЬНЫХ
ТЕХНОЛОГИЙ

Гарантия в течение 1 года!

КОМПЬЮТЕРНЫЙ КОМПЛЕКС
АРМ "АДМИНИСТРАТОР":
КОМПЬЮТЕР + ПРИНТЕР, ФАКС-МОДЕМ/ФАКС
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

СРЕДСТВА MULTIMEDIA:
ЗВУКОВЫЕ ПЛАТЫ
НАКОПИТЕЛИ CD-ROM
КОНТРОЛЛЕРЫ CD-ROM
МИКРОФОНЫ
НАУШНИКИ
КОМПАКТ ДИСКИ (CD)



ЛЮБАЯ КОМПЛЕКТАЦИЯ КЛАССОВ:
ОТ ЯДРА КЛАССА (МИНИ-КЛАСС)
ДО MULTIMEDIA-КЛАССА
МОДИФИКАЦИЯ КЛАССОВ

СТАНДАРТНЫЕ КЛАССЫ: 11 × 386SX/33	\$7,500
MULTIMEDIA КЛАССЫ: 486DX2/66 + 11 × 386SX/33	\$13,000

СЕТЕВОЕ ОБОРУДОВАНИЕ:
ПЛАТЫ ИОЛА, ETHERNET
РАЗВЕТВИТЕЛИ
КАБЕЛИ, РАЗЪЕМЫ
СЕТЕВОЕ МАТОБЕСПЕЧЕНИЕ

ОБУЧАЮЩИЕ И ДЕЛОВЫЕ
ПРОГРАММНЫЕ ПРОДУКТЫ
С СОПРОВОЖДЕНИЕМ И
ДОКУМЕНТАЦИЕЙ

Доставка по РОССИИ!

☎ (095) 535-2222, (095) 534-4831, факс (095) 534-4832

✉ 103498, Москва, К-498, а/я 164

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИИ
ЦЕНТР ИНТЕРАКТИВНЫХ СРЕДСТВ ОБУЧЕНИЯ

Новую жизнь классу **УКНЦ**

дадут

ВИНЧЕСТЕР

И

БЫСТРАЯ СЕТЬ



- ☑ Повысят надежность и ускорят работу в классе в 50 - 60 раз
- ☑ Избавят от работы с дисками
- ☑ Защитят данные от несанкционированного доступа

☑ Предложат удобный интерфейс

20 Мбайт винчестера Вы можете по своему желанию разделить на 8 устройств, с каждого из которых возможна загрузка своей операционной системы и любых программных средств.

По желанию клиента на винчестер может быть установлен программно-методический комплекс по базовому курсу информатики



УКНЦ

созданный в ходе образовательного проекта "Пилотные школы", а теперь доступный всем школам, оснащенным классами УКНЦ.

КОМПЛЕКС ВКЛЮЧАЕТ В СЕБЯ:

Учебник "Основы информатики и ВТ"	12 шт
Книга для ученика	12 шт
Книга для учителя	1 шт
Дискеты (80 дорожек)	2 шт

Цены до 15.03.1994:

комплекс "IBM на УКНЦ"	\$35
винчестер	\$205
быстрая сеть	\$510

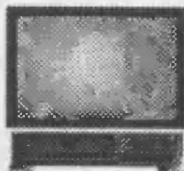
**Сборник
 видеофильмов**

" Основы

информатики

И

**ВЫЧИСЛИТЕЛЬНОЙ
 ТЕХНИКИ "**



- Основы теории информации
- Первоначальные сведения об информации
- Первоначальные

сведения об ЭВМ

- История развития вычислительной техники
- Устройство ЭВМ
- Компьютерные средства в учебном процессе

Сборник поставляется на 90 минутной кассете VHS

Цена сборника, включая НДС \$15

*По желанию заказчика кассета может быть
 выслана почтой*

Стоимость пересылки без возмещения
 убытков при пропоре бандероли 500 рублей

с возмещением 10000 рублей 2000 рублей

Оплата в рублях по текущему курсу
 ММДБ

РЕКВИЗИТЫ ЦЕНТРА

Адрес: 125315, г. Москва, ул. Часовая, 21
**Банковские реквизиты для Москвы и
 Московской обл. р/с 1609325 в Ленинградском
 отделении МББ, МФО 201694.**

Для других регионов: кор. счет 48616100
 в РКЦ ГУ ЦБ РФ, МФО 201791

Телефон: (095) 155 87 37, 155 87 30

Телефакс: (095) 155 87 27

КЛУБ «КОРВЕТ»

С. А. Никитин, г. Осташков

СЕКРЕТЫ «КОРВЕТА»

Система прерываний

Кого и зачем нужно прерывать? Вообразите себе большого Начальника, который сидит в своем кабинете и решает разные дела. Так как у Начальника, пусть даже большого, имеется только одна голова и он не может одновременно разбираться больше, чем с одним делом, у него есть Секретарша, которая выясняет, что за посетители приходят, а их сообщения сортирует в порядке важности для Начальника.

Конечно же, периодически Начальник говорит своей Секретарше, какие дела на текущий момент самые важные, какие могут подождать, а какие и вовсе недостойны отвлекать его внимание.

Может так случиться, что дело попало сверхсрочное и в то же время малозависящее от других текущих проблем. Тогда Начальник вызывает Секретаршу и приказывает ей никаких посетителей не впускать, по телефону не соединять. Теперь можно спокойно, не прерываясь по пустякам, решать свое сверхважное дело.

Но в другой раз дело оказалось таким, что без постоянной связи с внешним миром его не решить: все время надо реагировать на изменяющуюся обстановку. Начальник приказывает Секретарше: по такому-то поводу впускать и соединять немедленно, по такому — пусть в очереди сидят, а остальным — не смей прерывать мою работу!

Теперь назовем действующие лица своими именами: большой Начальник — это микропроцессор, Секретарша, как многие уже догадались, — контроллер прерываний, а посетители и телефонные звонки — не что иное, как запросы прерываний. Изложим подобную же историю в терминах компьютерной технологии.

Представьте себе, что программа работает с несколькими периферийными устройствами, например поддерживает обмен с другими машинами через локальную сеть и выводит какие-то данные на принтер. Если бы не было возможности прерывания, процессору пришлось бы постоянно опрашивать все эти устройства: нет ли запроса на передачу, нет ли готовности принять данные? Все равно что телефон без звонка: пока трубку не поднимешь, до тех пор не узнаешь — хочет ли кто передать сообщение, или линия свободна.

Очевидно, такая «обеспокоенность» процессора состоянием периферии была бы очень накладной и по времени, и по объему памяти, занимаемому опрашивающей программой. Поэтому разработчики микропроцессора мудро предусмотрели возможность подать на одну из соток его ножек сигнал прерывания.

Но, имея лишь один обезличенный сигнал прерывания, микропроцессору трудно разобраться — какое устройство его отвлекло и насколько важно для работы прерывание именно этого устройства для работы. Пришлось бы опять занудно опрашивать всех подчиненных на тему: «Кто сказал мяу?». А одновременное прерывание от нескольких устройств было бы вообще неразрешимой задачей.

Теперь вполне понятно, зачем микропроцессору понадобилась Секретарша в виде БИС ВН59, на которой и выполнен контроллер прерываний. Совершенно аналогичная микросхема под названием «8259» применяется в импортной аппаратуре, совместимой с IBM PC/XT/AT.

Пять «секретарских» обязанностей контроллера прерываний

Во-первых, контроллер прерываний увеличивает число независимых запросов прерывания. В ПК 80 используется только одна микросхема — Секретарша, поэтому

количество таких запросов ограничено восемью. В табл.1 перечислены источники запросов прерывания всех восьми уровней. Для любознательных добавим, что на каждый вход этой микросхемы можно подать выход точно такой же БИС. Это называется вторым каскадом. Несложно подсчитать, что число запросов при этом увеличилось бы до 15..64! Но конструкторы, видимо, не поленились прикинуть размеры электронной платы и стоимость машины при введении второго каскада прерываний и решили, что ежели «по-умному», то и восьми уровней вполне достаточно. С этим можно согласиться, но было бы очень полезно иметь еще и отсутствующее в данной машине прерывание от клавиатуры.

Во-вторых, контроллер прерываний сортирует запросы прерывания по задаваемым программно приоритетам. Это означает, что запрос, имеющий более высокий приоритет, обрабатывается в первую очередь. Такая ситуация возникает, когда два и более запроса приходят одновременно или очередной запрос приходит во время обработки предыдущего.

Приоритеты прерываний во многом сходны с приоритетами арифметических операций. Если в арифметическом выражении имеется более чем одна операция (например, сложение и деление), то сначала выполняется более приоритетная (в нашем примере — деление). Имеется даже аналог скобок, которые, как известно, повышают до максимума приоритет любой арифметической операции. Этот аналог будет описан чуть позже под названием «спецмаскирование». В табл.1 уровни прерываний перечислены по убыванию приоритета так, как они устанавливаются при стандартной инициализации (т.е. при включении или перезагрузке машины). Программист при необходимости может изменять приоритеты.

Таблица 1. Источники запросов прерывания

Уровень	Источник прерывания	Комментарии
0	Адаптер расширения системы	Любое внешнее устройство, подключенное к этому адаптеру, может генерировать запрос прерывания
1	Последовательный адаптер	Прерывание по окончании приема адаптером очередного байта
2	Последовательный адаптер	Прерывание по окончании передачи адаптером очередного байта
3	Адаптер локальной сети	Прерывание, свидетельствующее, что по сети пришел очередной байт
4	Генератор кадровой развертки	Прерывание с периодом 20 миллисекунд (т.е. 50 раз в секунду)
5	Канал 2 таймера	Прерывание с периодом, заданным программистом
6	Адаптер принтера	Прерывание, свидетельствующее, что принтер готов принимать данные
7	Контроллер НГМД	Прерывание, свидетельствующее о выключении мотора НГМД

В-третьих, контроллер прерываний может задерживать (не пропускать к процессору) прерывания любых уровней, независимо от их приоритета. Задержка контроллером одновременно всех уровней прерываний аналогична команде микропроцессора DI, но и в том и в другом случае контроллер-Секретарь хранит информацию о том, было ли обращение и от какого устройства, о чем его можно в любое время спросить.

Когда ПК 80 включают в сеть, программа инициализации ППЗУ программирует контроллер прерываний таким образом, что прерывания всех уровней, кроме третьего, задерживаются. Следовательно, только сообщение, пришедшее по локальной сети, может оторвать микропроцессор от работы. Это и понятно, ведь встроенные в ППЗУ программы могут обработать только прерывание от этого адаптера. Программы, использующие в своей работе другие устройства, должны корректно перепрограммировать контроллер (что и делает операционная система, если она сразу загружается с диска).

В-четвертых, нельзя было бы сравнить контроллер прерываний с секретарем, если бы он не умел провести без очереди «нужное» прерывание. Увы, и микропроцессор грешен, ибо может приказать контроллеру «впустить» любое прерывание, даже когда

более приоритетное еще не обработано. Называется такое «прерывание по благу» бюрократически скромно – спецмаскирование.

И наконец, в-пятых, в любой момент контроллер-Секретарша может предоставить микропроцессору отчет о состоянии своих внутренних регистров. Называется это режимом чтения и опроса. Подобный «маленький стриптиз» позволяет процессору узнать, были ли прерывания и какое из них имело наивысший приоритет в то время, когда прерывания программно запрещены (командой DI). Одно из применений режима чтения – это превращение входа прерывания в обыкновенный входной порт (ниже описано, как это используется на практике в базовых программах ПЗУ).

Что значит – обработать прерывание?

Конечно, «решения» по запросу должен принимать микропроцессор, это ведь он Начальник. Но привести его к правильному решению – задача чисто секретарская.

При поступлении запроса или запросов контроллер прерываний сначала решает две проблемы – «можно ли пропустить» и «кого первым», а затем сообщает процессору о том, что запрос «ждет». Микропроцессор может отреагировать сразу, закончив очередную команду, а может (в случае действия команды DI) – только после разрешения прерываний командой EI. «Услышав» контроллер, первым делом он выдаст ему сигнал подтверждения: «вас слышу». Только теперь контроллер прерываний посылает процессору три байта кодов – обычную команду CALL (OCDH) с адресом нужной подпрограммы.

От программиста этот «диалог» совершенно скрыт (он происходит на уровне «железа»), но его надо иметь в виду, иначе программа не будет правильно работать в реальном времени, особенно со «скоропортящимися» запросами (т.е. с запросами, требующими безотлагательного реагирования).

Дальше идут более очевидные события: микропроцессор выполняет подпрограмму, указанную контроллером прерываний, и, возвратившись из нее (как из обычной подпрограммы – по команде RET), продолжает прерванную работу. В конце подпрограммы обязательно должны быть команды, сообщающие контроллеру прерываний о том, что запрос обработан и его можно «снять с повестки дня». Если этого не сделать, контроллер будет считать, что процессор все еще занят обработкой запроса, и не пропустит к нему другие прерывания равного или более низкого уровня.

Кроме того, объявляя об окончании работы с запросом, процессор может одновременно сменить приоритеты уровней, но программы, манипулирующие приоритетами – это уже высший пил.таж и для ПК 80 таковых пока не встречается.

Наконец, откуда контроллер-Секретарь знает адреса нужных для обработки подпрограмм? Ему об этом заранее сообщает операционная система или прикладная программа – при инициализации (перезапуске) системы прерываний микропроцессор записывает в нужный регистр контроллера базовый адрес и формат таблицы переходов по запросам. Теперь контроллеру остается лишь прибавить к базовому адресу таблицы произведение номера запроса на ее формат (4 или 8) и выдать полученный адрес вслед за командой CALL. Не менее важным действием, которое должна заранее сделать программа, является заполнение указанной таблицы командами перехода согласно имеющимся процедурам обработки.

Этими действиями исчерпывается работа контроллера и микропроцессора по обслуживанию запроса прерывания. Пришла пора рассмотреть другую сторону системы – источники запросов.

Краткий обзор источников прерывания

Самыми традиционными источниками сигналов прерывания являются медленные внешние устройства и системный таймер.

Подавляющее большинство внешних устройств работает в сотни и тысячи раз медленнее, чем микропроцессор. Поэтому иногда (конечно же, не обязательно в каждом случае) имеет смысл не тратить попусту процессорное время, дожидаясь в цикле готовности того или иного адаптера, а подготовить подпрограмму, выполняющую нужные действия, разрешить прохождение сигналов прерывания соответствующего уровня и после этого спокойно заниматься текущими делами. При необходимости

устройство прервет процессор, который, уделив сотню-другую микросекунд на обслуживание запроса, снова будет свободен, порой на время, во много раз большее.

Уровень 0. Когда вы еще только достали машину из коробки, про уровень 0 можно сказать лишь то, что один из штырьков большого разъема на левом боку системного блока соединен с входной ножкой микросхемы контроллера прерываний. Число и функции устройств, которые могут пользоваться входом этого прерывания, ограничены только фантазией разработчика подключаемой аппаратуры. Естественно, что любая подобная конструкция должна быть поддержана программно, и запрос прерывания, если он разрешен, должен иметь соответствующую процедуру обслуживания. Принципы ее разработки будут изложены далее.

Зашитая в ПЗУ программа инициализации компьютера вообще использует вход 0 уровня как обычный входной порт. Дело в том, что, по соглашению, если используется внешнее ПЗУ, подключаемое к портам расширения системы, то она узнает об этом, проверяя, имеется ли перемычка между битом CONTROL и входом прерывания 0 уровня. Для этого программа, запретив прерывания, устанавливает бит CONTROL и в режиме чтения анализирует, есть ли запрос уровня 0.

Уровни 1 и 2. Эти прерывания генерируют, соответственно, приемник и передатчик последовательного адаптера.

Во время приема через последовательный канал адаптер сигнализирует микропроцессору о том, что в его входной регистр поступил и подготовлен для считывания очередной байт данных. Передача — процесс хотя и более предсказуемый, но столь же медленный. Ведь после записи в регистр данных адаптера, которая занимает считанные микросекунды, байт передается на линию в тысячи раз дольше. Поэтому можно воспользоваться запросом прерывания, который формируется адаптером сразу после окончания передачи байта. В этом случае запрос свидетельствует, что выходной регистр данных пуст и готов к записи следующей информации.

Уровень 3. К этому входу подключен приемник локальной сети. Поскольку адаптер ЛС является не чем иным, как немного упрощенным последовательным адаптером, это прерывание должно истолковываться совершенно аналогично 1 уровню: по локальной сети пришел и подготовлен к считыванию очередной байт информации.

Уровень 4. Источник этого прерывания — видеоконтроллер. Ход электронного луча в кинескопе дисплея управляется компьютером. Каждые 20 миллисекунд (или 50 раз в секунду) генератор кадровой развертки начинает очередной ход луча по экрану из левого верхнего угла. Иногда это называют «выводом видеобланка». Перед самым началом этого процесса видеоконтроллер посылает контроллеру прерываний свой сигнал.

Как ни покажется странным для новичка, дисплейное прерывание, пожалуй, самое ходовое в ПК 80. Прерывание 4 уровня используется не столько по прямому назначению — для согласования вывода на экран с частотой кадров, сколько в качестве фиксированного системного прерывания. Большинство операционных систем (CP/M, МикроDOS, Корнет), обрабатывая этот запрос, опрашивают клавиатуру. Это и есть цена отсутствия у контроллера клавиатуры собственной линии прерывания.

Уровень 5. Источник запроса — канал 2 микросхемы таймера. Говоря официальным языком, это прерывание системного таймера, но, поскольку таймер не умеет делать ничего, кроме отсчета временных интервалов, можно назвать этот запрос прерывания проще и понятней — «будильник».

В отличие от видеоконтроллера (запрос 4 уровня), который, как часы с кукушкой, посылает свой сигнал всегда через одно и то же время, таймер может по заданию микропроцессора отмерять различные интервалы. Сосчитав до нуля, он вежливо сообщает контроллеру прерываний: «заданное время истекло, извольте разбудить Начальника». Разбуженный процессор «идет» в соответствующую процедуру и делает работу, которая была подготовлена для выполнения именно в этот момент времени.

Уровень 6. Прерывание от принтера. Если вы знакомы с работой адаптера печатающего устройства, то никаких неясностей быть не может, так как данный запрос напрямую связан с сигналом BUSYP.

Программа может циклически опрашивать выходной порт принтера, выясняя состояние бита занятости, но иногда гораздо элегантнее указать в таблице переходов по прерываниям на процедуру вывода байта на печать, разрешить прохождение запросов 6 уровня и спокойно выполнять текущие дела, которых у микропроцессора немало. Такая печать имеет особое название — фоновая.

Низкий приоритет прерываний принтера объясняется тем, что печать — один из самых медленных способов вывода информации, а главное — запросы принтера «непортящиеся», т.е. при отсутствии очередного байта печатающее устройство может сколь угодно долго ожидать информации — на качество печатаемого текста это никак не повлияет.

Уровень 7. Редко используемое прерывание от контроллера НГМД. Этот контроллер имеет схему (одновибратор), которая включает мотор дисководов на 3 секунды после установки в единицу бита 5 в регистре управления. Так вот, инвертированный сигнал этого самого одновибратора и является запросом прерывания 7 уровня. Таким образом, запрос поступает в тот момент, когда выключается мотор дисководов. При включении же мотора запрос сбрасывается. Чуть позже будет рассмотрена программа на Бейсике-ПЗУ, иллюстрирующая прерывание этого уровня.

Единственное известное автору применение этого запроса не имеет никакого отношения к прерыванию. С его помощью программа инициализации ПЗУ проверяет наличие контроллера НГМД в данной машине, а следовательно, надо ли передавать управление загрузчику ОС или сразу перейти в Бейсик-ПЗУ. Этот способ не показывает наличие или отсутствия дисководов (т.е. сам НГМД может и отсутствовать), а говорит лишь о наличии контроллера. Зато это хороший прием, позволяющий программно отличить ПК 8020 от 8010.

Программирование системы прерываний

Высокий уровень. Увы, языки высокого уровня, как правило, не имеют собственного выхода на систему прерываний. Имеется в виду, что если оттранслированная программа и управляет прерываниями, то это результат работы транслятора, а не воля программиста, который для этого не имеет никаких явных средств, например операторов или функций.

Однако, изучив программирование системы прерываний на низком уровне, легко повторить то же самое и на высоком. Даже в Бейсике с помощью РЕЕК и РОКЕ можно сделать любую работу по программированию микросхемы контроллера. Но не торопитесь воспользоваться этим, не имея уверенности, что система программирования не изменит ваших установок. Если возникла необходимость манипулировать прерываниями, лучше всего создать подпрограмму на ассемблере, а потом «подшить» ее к откомпилированной основной программе. В системе Паскаль МТ+ для этого будет удобен встроенный миниассемблер, при этом полезно создать небольшую библиотеку процедур, программирующих систему прерываний.

Почти не найдется причин для того, чтобы, программируя на Бейсике, ломать себе голову о прерываниях, но один полезный совет для пользователей компьютеров, связанных локальной сетью, тут бесспорен. Чтобы защититься от прерываний по локальной сети, которые могут случайно нарушить работу Бейсика-ПЗУ, необходимо дать директиву РОКЕ &HFB29,&HFF. Тогда контроллер не пропустит запросов к процессору, и можно работать спокойно. Для восстановления «сетевго слуха» директива такая: РОКЕ &HFB29,&HF7. Этими же командами полезно начинать и заканчивать большинство Бейсик-программ, если недопустим их случайный сбой. Бейсик указанную установку никогда не отменяет, чего нельзя сказать, если вы попытаетесь запретить прерывания командой микропроцессора DI.

Средний уровень. Ни ПЗУ, ни операционные системы типа CP/M не имеют каких-либо доступных пользователю функций, программирующих контроллер прерываний. Но это чаще всего и не нужно. Важно, что они инициализируют систему прерываний оптималь-

ным образом и создают таблицы переходов по запросам. Для подавляющего большинства случаев этого достаточно. Надо лишь разобраться в том, как пользоваться этими таблицами, что совершенно несложно.

Таблица стартовых переходов по прерываниям. ПЗУ и операционная система создают совершенно одинаковые, полностью совместимые таблицы стартовых переходов по прерываниям. Такая «типовая» таблица всегда находится в самом конце системной страницы по фиксированному адресу F7E0H и имеет формат 4*8 байт. Структура ее такова: три байта под команду перехода, четвертый «пустой». Пример:

```

    . . . . .
0F7XX: JMP addr
      DS 1
    . . . . .

```

Такая последовательность команд может повторяться до 8 раз, хотя обычно большая часть таблицы пустует. Просмотреть реальную таблицу в своей системе можно либо с помощью программы типа DDT (SID), либо написав простенькую программку на Бейсике.

```

10 CLS: PCLS: PRINT TAB(16)"Таблица прерываний": PRINT
20 DIM D(3)
30 FOR I=0 TO 7 ' Для каждой строки таблицы
40 LOCATE 12,I+3: PRINT "0F7";HEX$(&HE0+I*4);": ";
50 FOR B=0 TO 3 ' Для каждого байта строки
60 D(B)=PEEK(&HF7E0+I*4+B)
70 NEXT B ' Следующий байт
80 IF D(0)=&HC3 THEN PRINT"JMP ";HEX$(D(2)*256+D(1));GOTO110
90 PRINT "DB ";
100 FOR B=0 TO 2:PRINTHEX$(D(B));", ";:NEXT:PRINTHEX$(D(3));
110 PRINT TAB(42) "; Уровень";I
120 NEXT I ' Следующая строка

```

В DDT нужно дать команду LF7E0,F7FF [BK]. Но здесь таблица получится не такая наглядная, так как деассемблер DDT «не знает» ее структуры и воспринимает пустые строки как последовательность команд (скорее всего, вы увидите NOP или RST 7). Теперь вы воочию убедились, что исследуемая система поддерживает всего одно-два прерывания.

В случае Бейсика-ПЗУ – это локальная сеть. ПЗУ, кроме программы инициализации, никаких программно доступных функций, программирующих контроллер, не имеет. Но ПЗУ имеет монитор локальной сети (тесно связанный с Бейсиком), запускающийся по запросу прерывания 3 уровня. Другие уровни прерываний программами ПЗУ не поддерживаются (если не считать загрузчика из внешнего ПЗУ, который использует запрос 0 уровня как дополнительный входной порт, о чем уже говорилось).

Стандартные версии операционных систем CP/M и МикроDOS поддерживают только прерывание, которое исходит от видеоконтроллера 50 раз в секунду. По каждому запросу ОС обрабатывает изменения в таблице цвета (LUT) и опрашивает клавиатуру. Сетевые версии имеют также процедуру, обслуживающую запрос локальной сети. Запрос системного таймера используется очень редко.

Таблица векторов прерываний. В отличие от ПЗУ, системная страница, организуемая ОС CP/M, кроме описанной, имеет таблицу векторов перехода. Откуда она взялась и зачем нужна?

Дело в том, что операционная система выполняет простейшую предобработку прерываний. Адрес в рассмотренной выше таблице переходов указывает на секцию команд этой предобработки. Обычно она сводится к запрещению прерываний, сохранению регистровых пар HL и PSW в стеке и переключению карты памяти в конфигурацию ODOSA (системная константа 1CH). Затем процедура передает управление подпрограмме, обрабатывающей прерывание. Этот переход организован косвенно, по вектору из описываемой таблицы. Вот типичная предобработка i-того прерывания:

0F7XX:	...	;	Фрагмент таблицы переходов
	JMP INT_i	;	по прерыванию
	...	;	...
INT_i:	...	;	Секция команд предобработки
	DI	;	Запретить прерывания
	PUSH H	;	Сохранить в стеке HL
	PUSH PSW	;	Сохранить в стеке PSW
	MVI A,ODOSA	;	Нормализовать карту (1CH)
	STA SYSREG	;	памяти в ODOСА (0FA7FH)
	LHLD VTRAPi	;	Взять вектор перехода из таблицы
	PCHL	;	Перейти по вектору

Структура таблицы векторов прерываний проще, чем предыдущей. Это простая последовательность двухбайтовых слов. Но элементов в этой таблице не восемь, а двенадцать – 8 векторов аппаратных прерываний и 4 так называемых псевдопрерываний (или программных прерываний). Псевдопрерывания на самом деле никакие не прерывания. Просто так организуют переход к некоторым системным процедурам (в частности, опроса клавиатуры), которые по смыслу аналогичны аппаратным прерываниям. Важное отличие: в нужном месте программы всегда имеется явный вызов подпрограммы псевдопрерывания, тогда как подпрограмма, обрабатывающая аппаратное прерывание, ниоткуда явно не вызывается (процессор попадает в нее по аппаратным сигналам контроллера прерываний).

Вот как выглядит таблица векторов прерываний, базовый адрес которой также фиксирован на системной странице (0F7C8H):

VTRAP0:	DW	DOINT0	;	0F7C8	Вектор прерывания 0 уровня
VTRAP1:	DW	DOINT1	;	0F7CA	
VTRAP2:	DW	DOINT2	;	0F7CC	
VTRAP3:	DW	DOINT3	;	0F7CE
VTRAP4:	DW	DOINT4	;	0F7D0	
VTRAP5:	DW	DOINT5	;	0F7D2	
VTRAP6:	DW	DOINT6	;	0F7D4	
VTRAP7:	DW	DOINT7	;	0F7D6	Вектор прерывания 7 уровня
VTRAP8:	DW	AUX10	;	0F7D8	Вектор псевдопрерывания 8 уровня
VTRAP9:	DW	AUX11	;	0F7DA
VTRAP10:	DW	AUX12	;	0F7DC	
VTRAP11:	DW	AUX13	;	0F7DE	Вектор псевдопрерывания 11 уровня

Это изображение таблицы, так сказать, в общем виде. Что представляет собой таблица векторов в ваших конкретных системах, покажет следующая программа:

```

10 CLS: PRINT "Таблица векторов прерываний": PRINT
20 FOR A=&HF7C8 TO &HF7DF STEP 2 ' Для каждого элемента таблицы
30 PRINT HEX$(A);": "; ' вывести адрес
40 PRINT HEX$(PEEK(A+1)); ' вывести старший байт
50 PRINT HEX$(PEEK(A)) ' вывести младший байт
60 NEXT A ' следующий элемент
    
```

Не удивляйтесь, что в Бейсике-ПЗУ таблица пустая. Об этой особенности прерываний ПЗУ будет сказано немного позже. Теперь же, после знакомства с системными таблицами, можно рассмотреть самый важный в этом разделе вопрос – как организовать собственную реакцию на прерывание.

Как организовать собственную реакцию на прерывание. Если ваша программа будет выполняться в среде ОС CP/M (или совместимых с ней), то почти все действия сведутся к манипуляциям с адресами в таблице векторов. Возможны два способа подключения к прерываниям:

1. Ваша новая процедура выполняет какую-то специфическую работу, после чего управление передается существующей («старой») процедуре обработки прерывания.

2. Ваша новая процедура полностью заменяет существующую, вызывать которую либо уже нет смысла, либо недопустимо.

Принципиальная разница состоит в окончании вашей процедуры: в первом случае достаточно (восстановив регистры BC, DE и SP) просто перейти по запомненному «старому» вектору, все остальное сделает ОС при выходе из обработки. Во втором случае ваша процедура должна сама восстановить все регистры, сообщить контроллеру об окончании прерывания, восстановить конфигурацию памяти, разрешить прерывания и вернуться по команде RET. Вообще-то, возможен и третий вариант, вернее немного видоизмененный второй – по окончании обработки передать управление секции команд, завершающей прерывание. Но, во-первых, ваша программа будет «привязана» к конкретной версии ОС (малопривлекательная перспектива), а во-вторых, «овчинка выделки не стоит», так как легче повторить этот крохотный командный фрагмент в заменяющей процедуре.

Действия по замене вектора прерываний в обоих случаях одинаковы. Вот корректный алгоритм этой работы, иллюстрируемый на ассемблере, для прерывания номер *i*:

DI		; Запретить прерывания
LHLD	VTRAP0+2*i	; Прочитать вектор из таблицы
SHLD	COPVTR+2*i	; и запомнить его в переменной-копии
LXI	H,NEWVTi	; Вектор собственной процедуры
SHLD	VTRAP0+2*i	; записать в таблицу
EI		; Разрешить прерывания

Примечание. Выбранная ассемблерная реализация, конечно, далеко не единственно возможная, но к преимуществу этого способа следует отнести то, что задействована всего одна регистровая пара. Адресация переменной-копии подсказывает, что это таблица копий; в более простом случае, когда такая копия одна, достаточно было бы указать ассемблеру лишь имя COPVTR.

Теперь обсудим саму процедуру обработки запроса для обоих способов подключения к прерыванию. В первом случае (более простом и предпочтительном) корректными будут следующие действия:

NEWVTi: ; "Добавка" к процедуре обработки *i*-того запроса

DI		; Запретить прерывания
PUSH	...	; При необходимости сохранить
...		; регистры, кроме PSW и HL
...		; ...
...		; (тело процедуры)
...		; ...
POP	...	; Восстановить сохраненные регистры
LHLD	COPVTR+2*i	; Вспомнить "старый" вектор из копии
PCHL		; Перейти в "старую" процедуру

Главное достоинство этого варианта – полная независимость как от версии ОС, так и от аппаратных средств. Если все же необходимо полностью заменить процедуру обработки, то возвращаться придется самостоятельно. Вот проверенный вариант:

NEWVTi: ; Новая процедура обработки *i*-того запроса

DI		; Запретить прерывания
PUSH	...	; При необходимости сохранить
...		; регистры, кроме PSW и HL
...		; ...
...		; (тело процедуры)
...		; ...
POP	...	; Восстановить сохраненные регистры
LDA	SYSCOPY	; Восстановить из копии (0F703)
STA	SYSREG	; конфигурацию памяти (0FA7F)
MVI	A,EOI	; Сообщить контроллеру о том, (20H)

STA	EOIREG	; что прерывание обработано (0FB28)
POP	PSW	; Восстановить регистры,
POP	HL	; сохраненные в предобработке
EI		; Разрешить прерывания
RET		; Вернуться в подпрограмму

Строки, программирующие контроллер напрямую, будут подробно объяснены в следующем разделе.

Одной из самых важных причин для подключения к прерыванию является дополнение процедуры, обрабатывающей запрос 4 уровня. После обновления таблицы векторов, 50 раз в секунду (если не запрещены прерывания) ваша процедура будет получать управление. Таким способом можно, например, проверять нажатие каких-то клавиш, которые в вашей программе являются управляющими (подобно Ctrl-C в Бейсике или Ctrl-P в ОС CP/M), а также обновлять какую-то индикацию на экране (счет, время, координаты и т. п.) или показания системных часов. Например, ваша процедура может подсчитывать время, за которое пользователь не произвел никаких действий, и, если оно превысит какую-то критическую величину (минут, допустим, пять-десять), программа очищает экран, чтобы не изнашивались катоды и люминесцентное покрытие дисплея. Кстати, сделать это можно двумя способами: либо сохраняя информацию в ОЗУ и заполняя экран пробелами, либо, что более элегантно, перепрограммируя таблицу LUT и не трогая с места ни единого байта. Пример этот не взят с потолка — так работает Norton Commander, который выводит при этом изображение «звездного неба», т. е. редкие неяркие точки, чтобы пользователь не подумал, что дисплей сломался или выключен. Не видно причин не пользоваться этим приемом и в других программах — телевизор штука дорогая. Однако не забудьте предусмотреть восстановление картинки при первом же нажатии любой клавиши (это тоже должно быть заложено в «добавку» к процедуре).

Как отключиться от прерывания? Не менее важно корректно отключиться от прерывания, восстановив работоспособность системы в первоначальном виде. Здесь возможны два способа:

1. Если необходимость в отключении от прерывания может возникнуть только по окончании программы, то специально ничего делать не надо. Завершите программу «теплым стартом» (JMP 0), который полностью реинициализирует систему прерываний.

2. Если не исключено, что потребуется восстановление исходных векторов по ходу программы, то необходимо предусмотреть следующие несложные действия:

DI		; Запретить прерывания
LHLD	COPVTR+2*i	; Прочитать из копии первоначальный
SHLD	VTRAP0+2*i	; вектор и поместить его в таблицу
EI		; Разрешить прерывания

Ограничения на процедуру обработки прерывания. Способ подключения к прерыванию путем изменения вектора очень удобен в большинстве ситуаций, так как немалую часть работы берет на себя операционная система. Однако у всякой палки два конца — необходимо помнить, что традиционные операционные системы ПК 8020 весьма ограниченно работают с прерываниями. ОС CP/M, в частности, требует от процедуры обработки прерывания следующего:

1. Должны быть сохранены регистры DE, BC и SP. Как уже указывалось, HL и PSW сохраняет процедура предобработки. Не лишним будет предварительно убедиться с помощью деассемблера, что в вашей версии ОС дело обстоит именно так.

2. Внутри процедуры нельзя вызывать функции BDOS и BIOS.

3. Нужно запретить прерывания, так как в большинстве версий не разрешаются вложенные запросы.

Могут быть и еще какие-то ограничения, о чем необходимо справиться в описании конкретной версии, установленной на вашем компьютере.

Подключение к прерыванию в системе ПЗУ. Программы ПЗУ, как уже говорилось, тоже инициализируют систему прерываний и поддерживают запрос приемника локальной сети. Но системная страница, организуемая ПЗУ, не имеет таблицы векторов, поэтому работать надо прямо с адресами в таблице перехода по прерываниям. Естественно, что системная предобработка запросов в этом случае не проводится. Следует помнить, что шаг таблицы переходов — не 2 байта, а 4, причем адрес начинается со второго байта. В первый байт элемента таблицы следует всегда записывать код команды JMP (ОСЗН), так как вы уже могли убедиться, что система делает это только для одного уровня запроса, остальные — пусты.

Классикой подключения к прерываниям в Бейсике-ПЗУ является программа «BASIC Vocabulary» (BASVOC.COM и BASVOC.BIN). По содержанию это почти бесполезный и педагогически «серый» Бейсик-справочник. Но программистское решение весьма интересно, так как очень изящно используется страничная организация памяти ПК 80. Справочная система загружена в те страницы ОЗУ, которые перекрываются ПЗУ (а это целых 24 килобайта), поэтому пользователь работает в Бейсике-ПЗУ, не испытывая никаких ограничений. Но в старших адресах ОЗУ имеется маленькая процедура, опрашивающая клавиатуру и передающая управление справочной системе, если нажата клавиша [ПРФ/esc]. Как вы уже, наверное, догадались, вызывает эту процедуру прерывание 4-го уровня.

Эти идеи могут лечь в основу множества самых разных программ, когда требуется разместить данные или команды, не используя для этого ОЗУ Бейсика. Возможны и обратные варианты, например версия CP/M, которая временно «подключает» пользователя к Бейсику ПЗУ при нажатии какой-либо управляющей клавиши.

Коварства ПЗУ-Бейсика. Дело в том, что Бейсик-ПЗУ создавался без особых мыслей о расширении системы. Наверное, это было бы правильно, если бы школьные компьютеры совершенствовались мировыми темпами. Но поскольку мы «застряли на месте» и нет возможности оснастить все РМУ дисководом или массово применять кассеты ПЗУ, приходится программно совершенствовать то, что заложено в машину изначально.

Опыт показал, что Бейсик-ПЗУ не поддерживает на системной странице копии нечитаемых регистров (системный, цветовой, LUT). Но самое печальное, что при этом он даже не запрещает прерывания, переключая карту памяти! Следовательно, нельзя размещать процедуру, обрабатывающую прерывание выше адреса ОС000Н или ниже адреса 8000Н, иначе ваши коды будут перекрыты видеопамятью ГЗУ. Но не переживайте сильно — в basic-буфере остается еще около 20 килобайт, это немало. Выяснить, в какой конфигурации очутилась ваша процедура, можно, прочитав ячейку 6000Н. Обнаружив нестандартную конфигурацию, лучше немедленно сбросить запрос и РЕТИроваться, иначе вас подстерегают непредсказуемые неприятности. Ценой долгих экспериментов можно научиться что-то делать на фоне бейсиковского рисования, но закраска все равно всегда выводит процедуры из строя.

Другая хитрость заключается в том, что программа инициализации ПЗУ устанавливает контроллер прерываний для чтения таблицы переходов с адреса 2Е0Н, т. е. в ПЗУ, куда вы не сможете записать свои JMP'ы и адреса. Поэтому необходимо реинициализировать контроллер (как описано далее), но поместить таблицу не на привычную для CP/M системную страницу OF700Н, а опять же ниже ОС000Н.

И это еще не все: Бейсик читает клавиатуру после каждой команды, не используя для этого системное прерывание. Учитывая то, что интерпретатор «зашит» в ПЗУ, последнее практически означает, что программист не в силах устранить бейсиковский контроль за клавиатурой. Например, невозможно отключить клавишу [СТОП], чтобы защитить программу от неуклюжих действий новичка.

Если после всех этих «НО» вы не передумали иметь дело с прерываниями в ПЗУ-Бейсике, в добрый путь. Тем более что большинство «граблей», на которые можно наступить, вам показаны.

Обратные варианты (например, версия CP/M, которая временно «подключает» пользователя к Бейсику-ПЗУ при нажатии какой-либо управляющей клавиши), наверное, возможны, но затруднительны, так как требуют безукоризненного знания программного интерфейса интерпретатора. Однако соблазнительно, например, получить доступ к «защитым» арифметическим процедурам Бейсика, ведь в этом он силен.

Итак, на среднем уровне доступ к системе прерываний наиболее удобен для программиста. Однако для понимания «души» этой системы и уверенного программирования в любых ситуациях, на любом уровне, необходимо спуститься еще на одну ступеньку — к портам и регистрам контроллера прерываний.

Низкий уровень. Специальные команды процессора. Конечно, первое, что следует упомянуть в программировании системы прерываний на низком уровне, — это команды микропроцессора DI (запретить прерывания), EI (разрешить прерывания) и HLT (останов до прерывания). Все остальные действия возможны лишь путем программирования БИС контроллера прерываний VN59.

В семействе ПК 80 не используются так называемые немаскируемые прерывания, т. е. прерывания, которые нельзя так или иначе запретить. Такие запросы, как правило, применяются для реагирования на совершенно неотложные ситуации, например, падение напряжения питания или деление на ноль. Как всегда, об отсутствии какой-либо возможности можно пожалеть, но не следует забывать и о том, что немаскируемые прерывания — всегда клубок сложных программных противоречий. В нашем же компьютере этих проблем никогда не возникает, ибо есть возможность полностью запретить все прерывания.

Общим правилом при использовании команд DI и EI является сведение к минимуму времени, когда прерывания запрещены. Не менее важен анализ фрагментов программ, выход из которых возможен только по прерыванию. Конечно, бывают случаи, когда программисты преднамеренно организуют «зависание» системы (например, в CP/M v. 2. 2 имеется такой фокус), но чаще это происходит вопреки чьей-то воле. Поэтому надо крайне осмотрительно пользоваться командой HLT и отказаться от структуры «вечного» цикла.

Программирование БИС контроллера прерываний. Микросхема контроллера представлена в адресном пространстве процессора двумя байтовыми портами. Их относительные адреса идут подряд: 28H и 29H. Старший байт адреса, как обычно, определяется конфигурацией памяти, и для самых «ходовых» из них — BASIC и ODOSA — абсолютные адреса равны OFB28H и OFB29H. На самом деле контроллер имеет регистров больше, чем два байта, но доступ к ним определяется, во-первых, очередностью обращения, а во-вторых, состоянием битов-идентификаторов.

Когда мы говорим об очередности обращения, мы имеем в виду то, что порты по разному воспринимают данные, записанные в них при инициализации и после нее. Биты-идентификаторы — это разряды 4 и 3 порта 28H, сообщающие контроллеру наши желания: хотим ли мы переустановить контроллер, или даем текущую команду, а если команду, то какого типа.

Режим инициализации. Логично начать с этого режима, хотя для программы это всегда будет переустановка, ведь при включении машины в сеть программа ПЗУ уже инициализировала микросхему.

Начинать надо всегда с записи в порт 28H, при этом бит 4 должен быть установлен — это и есть признак инициализации. Следом записывается байт в порт 29H. При этом контроллер понимает наши данные так, как это показано на рис. 1.

Единичка в разряде 1 порта 28H сообщает контроллеру, что мы используем только одну микросхему, а не каскад. Естественно, что в ПК 80 этот бит всегда будет именно таким.

Надеюсь, понятно, почему для адреса таблицы переходов контроллер требует всего 11 разрядов (A15—A5). Конечно же потому, что младшие биты адреса начала таблицы

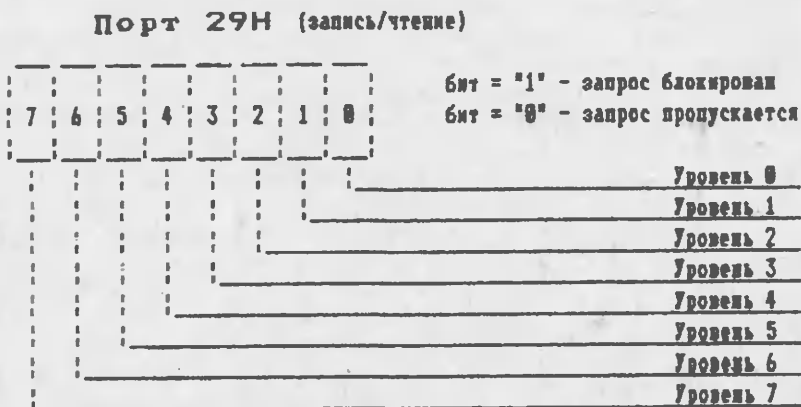


Рис.2. Регистр маскирования

не применяются, хотя могут очень элегантно решать некоторые программистские противоречия.

Обычное маскирование запросов

В отличие от тотального запрета прерываний микропроцессором, контроллер прерываний позволяет избирательно пропускать или задерживать любой запрос.

После записи двух байтов инициализации порт 29H контроллера воспринимает любую записанную в него информацию как маску блокируемых запросов и называется поэтому регистром маскирования (PM). Этот регистр произвольно доступен как для записи, так и для чтения. Замаскированный запрос скрыт от процессора. Восемь уровней прерываний соответствуют восьми разрядам регистра. Если бит установлен — прерывание данного уровня задерживается контроллером. Если бит обнулен — прерывание пропускается к микропроцессору (с учетом приоритета).

Запись в регистр маскирования — первое действие после инициализации контроллера. Причем сделать это надо до разрешения прерываний командой EI, так как регистр маскирования вначале обнулен и к процессору могут пройти нежелательные прерывания. В дальнейшем можно блокировать таким образом различные уровни запросов сколь угодно часто и в любых сочетаниях. Также ничем не ограничена возможность чтения регистра. Благодаря возможности простого чтения всегда предпочтительнее работать только с необходимыми в данный момент разрядами регистра, не изменяя остальные. Как это делается, подробно изложено в описании логических операций микропроцессора.

Для иллюстрации задайте Бейсику-ПЗУ следующую программу:

```

10 CLS: PRINT "Исходное состояние PM: ";
30 GOSUB 100 ' Вывод состояния регистра
40 PRINT TAB(7) "Смена маски: "; POKE &HFB29,&B11110000
50 GOSUB 100: PRINT "А сейчас система зависнет..."
60 POKE &HFB29,0: PRINT "Эта строка никогда не напечатается..."
70 END
100 PRINT BIN$(PEEK(&HFB29)): BEEP: PRINT: RETURN

```

Почему возникло зависание? Записав в регистр маскирования маску «0», мы решили прохождение к микропроцессору всех запросов. Однако в предыдущих параграфах было показано, что система ПЗУ не имеет ни векторов, ни процедур обработки для большинства прерываний. Напротив, ОС CP/M v. 2. 2 имеет все адреса переходов, а также пред- и постобработку (хотя сами процедуры обработки чаще всего отсутствуют). Благодаря этому в CP/M безопасно обнулять регистр маскирования, хотя и нежелательно.

Крохотную ассемблерную программку из четырех команд можно набрать, работая с утилитами DDT.COM или SID.COM. Вот протокол работы, отображаемый на экране дисплея (справа – пояснения):

A>DDT		Вызываем утилиту
DDT VERS	1.4	Заставка и промпт (#) DDT
#A100		Вставить ассемблерные
0100 EI		команды с адреса 100H
0101 MVI	A,0	Записать 0
0103 STA	FB29	в регистр маскирования
0106 JMP	0	"Теплый" старт ОС
0109		
#G100		Запуск фрагмента

Несмотря на то что эта программа записывает в регистр маскирования «0», т. е. пропускает все прерывания, зависания не происходит и мы благополучно возвращаемся в ОС. Напоминаю, что «теплый» старт полностью восстанавливает систему прерываний, в том числе и маски.

Специальное маскирование

Обрабатываемые запросы блокируют прохождение к процессору прерываний равного или низшего уровня. Обычный путь снятия этого блока – команда окончания прерывания. Но может возникнуть ситуация, когда необходимо пропустить заблокированный запрос, не заканчивая обработки предыдущего. В подобных случаях необходимо дать контроллеру команду специального маскирования (рис. 3г). В регистре маскирования при

Порт 28H (завись)

	7	6	5	4	3	2	1	0		
а	x	0	0	0	1	0	1	0	Режим чтения P03	0AH
б	x	0	0	0	1	0	1	1	Режим чтения P3	0BH
в	x	0	0	0	1	1	0	0	Режим сброса	0CH
г	x	1	1	0	1	0	0	0	Специальное маскирование	68H
д	x	1	0	0	1	0	0	0	Отмена спец. маскирования	48H

Рис. 3. Регистр ввода и спецмаскирования

этом устанавливают биты тех уровней, через которые должно «перепрыгнуть» пропускаемое прерывание. Команда 48H (рис. 3, д) отменяет режим спецмаскирования.

Чтение внутренних регистров

Контроллер предоставляет возможность прочитать свои внутренние регистры – регистр запросов (P3) и регистр обработанных запросов (P03).

Бит P3 устанавливается по приходу соответствующего сигнала от источника прерывания. О незамаскированном запросе с наивысшим приоритетом сообщается микропроцессору. Когда же тот подтвердит получение сигнала, бит из P3 перезаписывается в PO3, а в P3 сбрасывается. Теперь, пока процессор не сообщит об окончании обработки запроса, соответствующий бит PO3 будет установлен, а это блокирует другие равные или низшие запросы.

Читая P3 и PO3, программа может контролировать всю эту аппаратную «кухню» и при необходимости реагировать тем или иным образом. Например, программа инициализации ПЗУ выясняет, читая седьмой бит PO3, имеется ли в системе контроллер дисководов, и, если логическое включение мотора НГМД не обнуляет этот запрос, передает управление Бейсику-ПЗУ, минуя загрузчик ОС. Познакомьтесь с этим фрагментом, деассемблировав ПЗУ с адреса 048DH по 049CH.

Та же программа, читая PO3, проверяет наличие переключки между битом CONTROL порта 32H и входом прерывания 0 уровня. Бит CONTROL сначала обнуляется, а затем устанавливается. По реакции на эти переключения нулевого разряда PO3 делается вывод о наличии или отсутствии переключки, а следовательно, внешнего ПЗУ. Данный фрагмент можно просмотреть в адресах 04BFH...04D3H.

Если регистр маскирования имеет собственный порт и читается тривиально, то перед прочтением внутренних регистров необходимо дать контроллеру соответствующую команду. Команды, предшествующие чтению внутренних регистров, показаны на рис. 3, а, 3, б. Эти константы надо записать в порт 28H, а затем прочитать из него же.

Режимы чтения P3 и PO3 отменяются только подачей в порт 28H новой команды и не зависят от числа прочтений регистров.

Попробуйте прочитать PO3 контроллера в Бейсике-ПЗУ:

```
10 CLS: PRINT "Регистр маски:"; TAB(15); BIN$(PEEK(&HFB29))
20 POKE &HFB28,&H0A ' устанавливаем режим чтения PO3
30 PRINT "P O 3:"; TAB(15); BIN$(PEEK(&HFB28)): END
```

Полученные результаты следует понимать так: в обработке контроллером находятся запросы 4 и 7 уровней, но так, как эти уровни маскированы, в данный момент прерывания не доходят до процессора.

Следующая программа для Бейсика-ПЗУ ПК 8020 очень наглядно иллюстрирует запрос прерывания от одновибратора НГМД. Напомню, что это устройство включает мотор дисковода на 3 секунды, при этом сигнал запроса 7 уровня сбрасывается. Одновременно с выключением мотора запрос прерывания устанавливается. Подключен ли дисковод к компьютеру, для программы несущественно, так как она взаимодействует с контроллером НГМД, который встроен в системный блок и по таким «пустякам», как включение мотора, не проверяет состояние накопителя. Подробности, касающиеся НГМД, здесь не рассматриваются. Эта программа лишь позволяет в динамике наблюдать изменения 7 бита PO3 в зависимости от включения или выключения мотора НГМД.

```
10 CLS: PRINT "Иллюстрация к прерыванию от дисковода"
20 ROZ=&HFB28 ' Адрес PO3
30 RUD=&HFB39 ' Адрес регистра управления НГМД
40 POKE RUD,3 ' Сброс битов включения моторов
50 POKE ROZ,&H0A ' Режим чтения PO3
60 PRINT BIN$(PEEK(ROZ)) ' Чтение и печать PO3
70 POKE RUD,&H21 ' Включение на 3 с мотора НГМД А
80 FOR Z=0 TO 11
90 NR=PEEK(ROZ) ' Очередное прочтение PO3
100 T=8-LEN(BIN$(NR)) ' Вычисление табуляции для вывода
110 ? TAB(T); BIN$(NR) ' Вывод на экран
120 FOR Q=0 TO 300: NEXT ' Пауза
130 NEXT Z
140 END
```


Опрос на прерывание

У обычного режима работы системы прерываний – обслуживания по запросу – есть альтернатива: обслуживание по результатам опроса. В этом режиме запросы обрабатываются по инициативе программы, поэтому прерывания запрещены командой микропроцессора DI. Режим этот напоминает уже описанный выше телефон без звонка, однако в некоторых ситуациях он может быть полезен.

В определенные программистом моменты времени процессор опрашивает контроллер на предмет прерывания. Для этого в порт 28H записывается команда опроса (рис. 3, в). Теперь можно прочитать этот же порт. Полученный байт имеет структуру, изображенную на рис. 4.



Рис. 4 Структура данных при опросе на прерывание

Если седьмой разряд установлен, то в первых трех битах можно найти номер прерывания наивысшего уровня из запросивших. Если же седьмой разряд обнулен, анализ первых трех не имеет смысла, так как там заведомо находится 111В (а вовсе не нули, что может сбить с толку, если этого не знать).

В отличие от чтения P3 и P03, режим опроса действует только для однократного прочтения порта. Для того чтобы еще раз опросить контроллер, необходимо снова записать команду и лишь затем считать результат.

Когда целью опроса действительно является реакция на выявленный запрос (а не решение какой-то побочной проблемы), организуется переход в подпрограмму обработки. При этом номер прерывания, полученный в результате опроса, используется в качестве индекса для адресации на нужный вектор перехода.

Операции окончания прерываний

После обработки запроса необходимо сообщить контроллеру о том, что прерывание обработано. Контроллер при этом сбрасывает соответствующий бит во внутреннем регистре обслуженных запросов.

Если явно не указать об окончании прерывания, контроллер не пропустит к микропроцессору прерываний, низших или равных по приоритету обработанному. Обращение к регистру окончания прерываний (РОП) – через порт 28H. РОП выбирается сброшенными битами 4 и 3 (рис. 5).

Самый традиционный способ объявить прерывание окончанным – записать в РОП константу 20H, что указывает на обычное окончание и просто обнуляет соответствующий бит РОЗ (рис. 5, а). Пожалуй, это самая частая команда контроллеру, так как ею заканчиваются практически все процедуры обслуживания запросов.

Возможно и специальное окончание прерывания, когда обслуживается одно прерывание, а сбрасывается бит РОЗ другого уровня (рис. 5, б). При этом младшие три бита команды (B2–B0) указывают номер уровня сбрасываемого запроса. В обоих случаях приоритеты остаются неизменными, что чаще всего и требуется.

	7	6	5	4	3	2	1	0		
а	0	0	1	0	0	x	x	x	Обычное окончание	28H
б	0	1	1	0	0	B2	B1	B0	Специальное окончание	6BH+Num
в	1	0	1	0	0	x	x	x	Окончание с циклическим сдвигом приоритетного кольца	6AH
г	1	1	1	0	0	B2	B1	B0	Окончание с произвольным сдвигом приоритетного кольца	6EH+Num
д	1	1	0	0	0	B2	B1	B0	Установка приоритетного кольца без окончания	6CH+Num

Рис. 5 . Регистр окончания и сдвига приоритетов

Операции установки приоритетов

Ранее были перечислены источники прерываний в порядке убывания приоритетов. Такой порядок устанавливается по умолчанию, так как не требует никаких явных действий. Однако приоритеты запросов можно программно изменять. Это очень сильный прием, особенно в системах, работающих в реальном времени со специальными периферийными устройствами, но в стандартном ПК 80 ему трудно найти достойное применение, кроме как в играх, рассчитанных на двоих партнеров. Следует отметить, что приоритеты не могут меняться беспорядочно, так как относительное расположение входов прерываний задано аппаратно. Поэтому речь идет не о вольном манипулировании, а лишь о сдвигах приоритетного кольца. Причем все команды задают лишь низший уровень, остальные уровни определяются порядком расположения в кольце. Таким

а	б	в	г	д	е	ж	з	
0	1	2	3	4	5	6	7	<-высший приоритет
1	2	3	4	5	6	7	0	
2	3	4	5	6	7	0	1	
3	4	5	6	7	0	1	2	
4	5	6	7	0	1	2	3	
5	6	7	0	1	2	3	4	
6	7	0	1	2	3	4	5	
7	0	1	2	3	4	5	6	<-низший приоритет
<u>первичная автоустановка</u>								

Рис.6. Варианты распределения уровней приоритета

образом, возможны только восемь вариантов (рис. 6, а-з).

Контроллер ВН59 предоставляет три способа сдвинуть приоритетное кольцо. Перечислим их кратко:

1. Произвольный сдвиг приоритетов. В порт 28H записывается команда, изображенная на рис. 5, д, при этом разряды B2-B0 выбирают низший уровень приоритета.
2. Произвольный сдвиг приоритетов с окончанием прерывания (команда на рис.5,г). Ничем не отличается от предыдущего по действию на приоритеты, но одно-

временно осуществляет обычное окончание прерывания, сбрасывая в РОЗ бит обслуженного уровня.

3. Циклический сдвиг приоритетов с окончанием прерывания (рис. 5, в). Обычное окончание прерывания, при этом обработанный уровень автоматически попадает на дно приоритетного «колодца».

Мост к семейству IBM PC

Первым делом необходимо выяснить одну важную деталь. В большинстве руководств по IBM PC авторы относят к прерываниям и то, что мы выше называли аппаратными прерываниями, и так называемые программные или псевдопрерывания. Здесь, скорее всего, играет роль обычная психологическая инерция, когда неудачная (но устоявшаяся) терминология сводит «под одну крышу» принципиально различные понятия. Дело в том, что программные «прерывания» ничего не прерывают! Это просто специфический вызов подпрограмм. Отличие команды INT от команды CALL состоит лишь в том, что требуется указывать не физический адрес, а лишь номер (полная аналогия с командой RST для нашего микропроцессора). Но не более того. Нет главных признаков истинного прерывания – непредсказуемости и неявного вызова, ведь в процедуру обработки программного «прерывания» можно попасть только одним способом – явно вызвав ее. Аналогично организованы псевдопрерывания CP/M, о чем шла речь раньше, тот же принцип положен в метод «общей точки входа». Процедуры же, обслуживающие настоящие прерывания, ниоткуда не вызываются. Микропроцессор попадает в них принципиально другим способом – по аппаратным управляющим кодам контроллера прерываний.

Система обработки истинных аппаратных прерываний в PC решается абсолютно так же, как в ПК 80. Более того, на низком уровне программирования контроллера вы найдете отличия только в физической адресации двух портов микросхемы 8259. Остальное – совершенно аналогично. Но кроме обычных аппаратных микропроцессоры PC имеют еще два типа прерываний: внутренние и немаскируемые.

Системный таймер

Добрая хозяйка всегда одновременно готовит несколько блюд. Конечно, она может судить об их готовности методами «на глазок» и «по вкусу», но согласитесь, процесс станет более размеренным и цивилизованным, если на кухне появится маленький полезный прибор – таймер. Он подскажет своим сигналом о том, когда надо что-то выключить, а что-то посолить.

Микропроцессор – хозяйка не менее занятая. Ему приходится одновременно и клавиши опрашивать, и на экран выводить, и с диском обмениваться, и что-то печатать в фоновом режиме. И каждое устройство, и пользователь за дисплеем требуют совершенно различного временного подхода. Часть этих проблем решают прерывания непосредственно от устройств, но есть и такие, с которыми без хронометража не справиться.

Разговор о таймере совсем не случайно начат лишь после описания системы прерываний. Очень важно, чтобы не просто отсчитывалось время, но и была возможность оперативно узнать о том, что заданный интервал истек. Для этого в роли звонка будильника таймер использует собственный канал прерывания.

Иногда системный таймер называют «часами», но это не так. Английское слово «timer» действительно означает устройство, обрабатывающее время. Но часы-то по-английски называются «clock». Не стоило бы вдаваться в дебри лингвистики, если бы в них не был отражен важный для нас факт – со временем можно работать двумя альтернативными способами. Часы отсчитывают время по возрастающей и практически не предполагают окончания счета, ведь календарь, представляется как бы старшими разрядами времени, не имеет обозримого предела. Таймер же, напротив, считает по убывающей, и счет этот заведомо должен закончиться на нуле.

В схему компьютера, в принципе, могут входить и системные часы, но пока это принято лишь в более дорогих и сложных моделях, а в дешевых «персоналках» (в том числе и в ПК 80, и в IBM PC) часы делаются программно, кстати, на основе сигналов от системного таймера.

Таблица 2. Функции микропроцессора ПК «Корвет» (KP580BM80A)

Именованная команда	Выполняемое действие	Флаги		Именованная команда	Выполняемое действие	Флаги	
		FC	FAС!Dr			FC	Dr
Команды пересылки данных и обмена				Команды циклического сдвига			
Регистровая адресация				С прямой адресацией			
MOV Rg1, Rg2	Rg1 := Rg2 (нов. н.н. - шельза)	-	-	BLC	Сдвиг АКК влево	+	-
STAX RgP*	M(RgP) := АКК	-	-	RRC	Сдвиг АКК вправо	+	-
LDAX RgP*	АКК := M(RgP)	-	-	BAL	Сдвиг АКК влево через FC	+	-
XCBSG	HL := DL (обмен)	-	-	RAR	Сдвиг АКК вправо через FC	+	-
Прямая адресация				Команды перехода			
STA Adr	M(Adr) := АКК	-	-	С прямой адресацией			
LDA Adr	АКК := M(Adr)	-	-	JMP Adr	PC := Adr	-	-
SHLD Adr	M(Adr) := L, M(Adr+1) := H	-	-	J- Adr	Префикс условных переходов	-	-
LHLD Adr	L := M(Adr), H := M(Adr+1)	-	-	С косвенной адресацией			
Косвенная адресация				PCHL	PC := HL (регистровая)	-	-
MVI Rg, dB	Rg := dB	-	-	BT	PC := M(SP)_M(SP+1) (стековая)	-	-
LXI RgP, dW	RgP := dW	-	-	Команды вызова подпрограммы			
Операции со стеком				CALL Adr	M(SP-1)_M(SP-2) := PC,	-	-
PUSH RgP**	M(SP-1)_M(SP-2) := RgP, SP := SP-2	+	+	PC := Adr, SP := SP-2	-	-	
POP RgP**	RgP := M(SP)_M(SP+1), SP := SP+2	+	+	C- Adr	Префикс условных вызовов	-	-
SPHL	SP := HL	-	-	С адресацией по вектору			
XTBL	M(SP) := L, M(SP+1) := H	-	-	BT Num	M(SP-1)_M(SP-2) := PC, SP := SP-2	-	-
Ввод - вывод (в "Корвете" не применяются)				PC := 0000H+8*Num (Num=0..7)	-	-	-
IN dB	АКК := Порт с номером dB	-	-	Команды возврата из подпрограммы			
OUT dB	Порт с номером dB := АКК	-	-	RET	PC := M(SP)_M(SP+1), SP := SP+2	-	-
Арифметические команды				R-	Префикс условных возвратов	-	-
Сложение				Сuffиксы условий команд			
ADD Rg	АКК := АКК + Rg	+	+	-C	Если был перенос (FC = 1)	-	-
ADC Rg	АКК := АКК + Rg + FC	+	+	-NC	Если не было переноса (FC = 0)	-	-
ADI dB	АКК := АКК + dB	+	+	-Z	Если результат нулевой (Z = 1)	-	-
ACI dB	АКК := АКК + dB + FC	+	+	-NZ	Если результат ненулевой (Z = 0)	-	-
INR Rg	Rg := Rg + 1	-	-	-M	Если результат отрицательный (S = 1)	-	-
INX RgP	RgP := RgP + 1	-	-	-P	Если результат положительный (S = 0)	-	-
DAD RgP	HL := HL + RgP	+	-	-PE	Если результат четный (P = 1)	-	-
Коррекция (только после сложения)				-PO	Если результат нечетный (P = 0)	-	-
DAA	Десятичная коррекция АКК	+	+	Команды управления микропроцессором			
Вычитание				MOP	! Колостой код (пустая операция)	-	-
SUB Rg	АКК := АКК - Rg	+	+	Управление флагами			
SBB Rg	АКК := АКК - Rg - FC	+	+	STC	FC := 1 (установка FC)	+	-
SUI dB	АКК := АКК - dB	+	+	CMC	FC := NOT FC (инверсия FC)	+	-
SBI dB	АКК := АКК - dB - FC	+	+	Внешняя синхронизация			
DCR Rg	Rg := Rg - 1	-	+	HLT	Останов до запроса прерывания	-	-
BCX RgP	RgP := RgP - 1	-	-	HI	Разрешение прерываний	-	-
Сравнение				DI	Запрещение прерываний	-	-
CMP Rg	АКК - Rg --> изменение флагов	+	+	Условные обозначения			
CPI dB	АКК - dB --> изменение флагов	+	+	PC	- счетчик команд; SP - указатель стека	-	-
Логические команды				Rg	- регистр A, B, C, D, E, H, L или ячейка M(HL)	-	-
CMA	АКК := NOT АКК (инверсия)	0	-	RgP	- регистры пар BC, DE, HL, SP	-	-
ANA Rg	АКК := АКК AND Rg	0	?	RgP*	- регистры пар BC, DE	-	-
ORA Rg	АКК := АКК OR Rg	0	0	RgP**	- регистры пар BC, DE, HL, PSW	-	-
XRA Rg	АКК := АКК XOR Rg	0	0	Влияние на флаги: <←> - влияет; <-> - не влияет			
ANI dB	АКК := АКК AND dB	0	?	-	-	-	-
ORI dB	АКК := АКК OR dB	0	0	-	-	-	-
XRI dB	АКК := АКК XOR dB	0	0	-	-	-	-

Таблица 3. Коды команд микропроцессора «Корвет»

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	NOP	LXI BC, dW	STAX (BC)	INX BC	INR B	DCR B	NYI B, dB	RLC	-	DAD BC	LDAX (BC)	DCX BC	INR C	DCR C	NYI C, dB	RRC	0
1	-	LXI DE, dW	STAX (DE)	INX DE	INR D	DCR D	NYI D, dB	RAL	-	DAD DE	LDAX (DE)	DCX DE	INR E	DCR E	NYI E, dB	RAB	1
2	-	LXI HL, dB	SHLD Adr	INX HL	INR H	DCR H	NYI H, dB	DAA	-	DAD HL	LHLD Adr	DCX HL	INR L	DCR L	NYI L, dB	CMA	2
3	-	LXI SP, dW	STA Adr	INX SP	INR H	DCR H	NYI H, dB	STC	-	DAD HP	LDA Adr	DCX SP	INR A	DCR A	NYI A, dB	CMC	3
4	MOV B, B	MOV B, C	MOV B, D	MOV B, E	MOV B, H	MOV B, L	MOV B, M	MOV B, A	MOV C, B	MOV C, C	MOV C, D	MOV C, E	MOV C, H	MOV C, L	MOV C, M	MOV C, A	4
5	MOV D, B	MOV D, C	MOV D, D	MOV D, E	MOV D, H	MOV D, L	MOV D, M	MOV D, A	MOV E, B	MOV E, C	MOV E, D	MOV E, E	MOV E, H	MOV E, L	MOV E, M	MOV E, A	5
6	MOV H, B	MOV H, C	MOV H, D	MOV H, E	MOV H, H	MOV H, L	MOV H, M	MOV H, A	MOV L, B	MOV L, C	MOV L, D	MOV L, E	MOV L, H	MOV L, L	MOV L, M	MOV L, A	6
7	MOV M, B	MOV M, C	MOV M, D	MOV M, E	MOV M, H	MOV M, L	HLT	MOV M, A	MOV A, B	MOV A, C	MOV A, D	MOV A, E	MOV A, H	MOV A, L	MOV A, M	MOV A, A	7
8	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A	ADC B	ADC C	ADC D	ADC E	ADC H	ADC L	ADC M	ADC A	8
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB M	SUB A	SBB B	SBB C	SBB D	SBB E	SBB H	SBB L	SBB M	SBB A	9
A	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A	XRA B	XRA C	XRA D	XRA E	XRA H	XRA L	XRA M	XRA A	A
B	ORA B	ORA C	ORA D	ORA E	ORA H	ORA L	ORA M	ORA A	CMP B	CMP C	CMP D	CMP E	CMP H	CMP L	CMP M	CMP A	B
C	RNZ	POP BC	JNZ Adr	JMP Adr	CNZ Adr	PUSH BC	ADI dB	RST 0	RZ	RNY	JZ Adr	-	CZ Adr	CALL Adr	ACI dB	RST 1	C
D	RNC	POP DE	JNC Adr	OVI dB	CNC Adr	PUSH DE	SUI dB	RST 2	RC	-	JC Adr	IN dB	CC Adr	-	SBI dB	RST 3	D
E	RPO	POP HL	JPO Adr	XTBL	CPO Adr	PUSH HL	ANI dB	RST 4	RPK	PCBL	JPE Adr	XCHG	CFP Adr	-	XRI dB	RST 5	E
F	RP	POP PWH	JP Adr	DI	CFP Adr	PUSR PWH	ORI dB	RST 6	RK	SPBL	JM Adr	EI	CM Adr	-	CPI dB	RST 7	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Примечание. По вертикали берется старший HEX-разряд кода, по горизонтали — младший.

Регистровые пары обозначены полными именами (BC, DE, HL).

В квадратных скобках [.] указана косвенная регистровая адресация.

dB = 8-разрядный операнд; dW = 16-разрядный операнд; Adr = 16-разрядный адрес.

Таблица 4. Карта памяти страницы периферийных портов области УВВ

Микросхема	Адрес	Программируемый узел	Биты	Функция и доступность чтения-записи
КР5800В53	00H	Регистр счета канала 0	0-7	Выбор частоты сигнала звукогенератора
	01H	Регистр счета канала 1	0-7	Выбор скорости передачи через последовательный интерфейс
	02H	Регистр счета канала 2	0-7	Выбор задержки для генерации системных прерываний
	03H	Регистр режима таймера	0-7	Выбор способа чтения/загрузки байтов, режима БИС
КР5800В55А ЭПА-2	08H	Порт А	0-7	Порты А, В и С ЭПА-2 доступны только в распоряжение пользователя и выведены на разъем интерфейса расширения системы
	09H	Порт В	0-7	
	0AH	Порт С	0-7	
	0BH	Порт управляющего слова	0-7	
КР5800В51А	10H	Регистр данных	0-7	Объем данных через последовательный интерфейс
	11H	Регистр состояния	0-7	Чтение флагов квитирования и ошибок
	11H	Регистр управления	0-7	Запись команд адаптеру последовательного интерфейса
КР1818ВГ93	18H	Регистр команд	0-7	Запись команд контроллеру ГИМД
	18H	Регистр состояния	0-7	Чтение флагов квитирования и ошибок
	19H	Регистр дорожки	0-7	Чтение номера текущей дорожки ГИМД
	1AH	Регистр сектора	0-7	Запись номера искомого сектора ГИМД
	1BH	Регистр данных	0-7	Объем данных через контроллер ГИМД
КР5800В51А	20H	Регистр данных	0-7	Объем данных через адаптер локальной сети
	21H	Регистр управления	0-7	Запись команд адаптеру локальной сети
	21H	Регистр состояния	0-7	Чтение флагов квитирования и ошибок
КР5800В59	28H	Регистр окончания прерывания	0-7	0 - идентификатор регистра окончания прерывания Выбор конца прерывания и уровня низкого приоритета
	28H	Регистры опроса, чтения и спецпрерывания	0-7	0 - идентификатор регистров Чтение внутренних регистров ВИС 1 - выбор режима опроса 2 - управление спецпрерываниями 6-8
	29H	Регистр запрета прерывания	0-7	Биты устанавливаемые в '1' запрещают однократные уровни прерывания (доступны для записи и чтения)
КР5800В55А Порты А, В, С ЭПА-2	30H	Регистр данных принтера	0-7	Запись байтов данных для вывода на печать
	31H	---	0-7	Не используется (резерв)
	32H	Регистры входных сигналов периферийных устройств	0-1 2 3 4-6 7	Регистр записи адаптера ЭКИД Регистр управления историей ЭКИД ('1' - аквачен) Регистр разрешения звукогенератора ('1' - включен) Регистр управления адаптером принтера Бит CONTROL (аквачен на разъем расширения системы)
	33H	Порт управляющего слова	0-7	При специализации режим выбирается константой 82H
КР5800В55А Порты А, В, С ЭПА-1	38H	Регистры входных сигналов от периферийных устройств	0 2 3 4-7	Регистр чтения адаптера ЭКИД Регистр чтения состояния принтера Регистр чтения состояния трибуна АРЗУ Регистр чтения номера рабочего места
	39H	Регистр выбора и управления ГИМД	0-3 4 5 6 7	Выбор дисководов ('0001' - А, '0010' - В) Выбор стороны диска Запуск дисководов на 3 секунды ('0' - '1') Выбор плотности записи ('0' - двойная) Выбор диаметра диска ('0' - пятисантиметровый)
	3AH	Регистры управления отображением и выбора экранов ГЗУ	0-1 2 3 4-5 6-7	Выбор экрана ГЗУ (0-3), отображаемого дисплеем Выбор энергогенератора ('0' - осциллой) Выбор ширины символа ('0' - нормальная) Управление битом инверсии Выбор экрана ГЗУ (0-3), подымаемого к процессору
	3BH	Порт управляющего слова	0-7	При специализации режим выбирается константой 90H

Таблица 5. Карта конфигураций доступной микропроцессору памяти

Имя конфигурации и константы системного регистра		Адреса ПЗУ	Адреса ОЗУ	Адреса ГЗУ	Адреса УВВ	Область УВВ			
						Страницы клавиатуры	Страница служебных регистров	Страница периферийных регистров	Страницы АЦЗУ
YRS_80	80H	0-3FFF	4000-FFFF	---	3800-3FFF	3900	3A00	3B00	3C00
NDOS	14H	0-1FFF	2000-FFFF	---	7800-FFFF	F000	F400	F800	FC00
BOMB_1	16H	0-3FFF	4000-FFFF	---	F800-FFFF	F000	F400	F800	FC00
ODOSA	1CH	---	0-FFFF	---	F800-FFFF	F000	F400	F800	FC00
BOMB_2	28H	0-3FFF	4000-BFFF	C000-FFFF	3800-3FFF	3000	3A00	3B00	3C00
DOSG_1	3CH	---	0-3FFF и 8000-FDFF	4000-FFFF	F800-FFFF	---	F000	F800	---
BASIC	48H	0-5FFF	6000-FFFF	---	F800-FFFF	F000	F400	F800	FC00
DOSA	5CH	---	0-FDFF	---	F800-FFFF	---	F000	F800	---
BASG	60H	0-5FFF	6000-BFFF	C000-FFFF	B700-BFFF	---	B700	---	---
DOSG_2	6CH	---	0-BFFF	C000-FFFF	B700-BFFF	---	B700	---	---

- Примечание:**
1. Конфигурации перечислены в порядке возрастания константы системного регистра.
 2. Все адреса даны в HEX-форме.
 3. Для страниц области УВВ показаны только начальные (базовые) адреса.
 4. Базовые конфигурации: BASIC - для Бэйсика ППЗУ;
DOSA - для операционной системы МикроDOS;
ODOSA - для операционной системы CP/M-80.
 5. Сокращения: ОЗУ - оперативное запоминающее устройство;
ППЗУ - постоянное перепрограммируемое запоминающее устройство;
ГЗУ - графическое запоминающее устройство;
УВВ - устройство ввода-вывода.

КЛУБ УКНЦ

В. А. Брусенцев, А. В. Козин,

Республиканский Центр интерактивных средств обучения Министерства образования РФ

IBM на УКНЦ

Качественно новый программно-методический комплекс по информатике для КУВТ УКНЦ, 6-7 классы средней школы

В ходе экспериментального образовательного проекта «Пилотные школы», проводимого совместно с фирмой IBM, был создан, апробирован и внедрен в более чем 1000 школ бывшего СССР новый курс информатики.

Предлагаемый программно-методический комплекс (ПМК) делает этот курс доступным огромному числу школ, оснащенных классами УКНЦ.

Комплекс методически и программно идентичен своему аналогу на IBM и поддерживает следующие разделы курса:

- Основы алгоритмизации (полная поддержка глав 2—5 учебника «Основы информатики и вычислительной техники» А.Г. Гейн и др.);
- Организация данных (поддержка главы 6);
- Информатика — отрасль производства (поддержка главы 9).

Для быстрой и успешной организации учебного процесса ПМК укомплектован следующими пособиями:

- учебник «Основы информатики и ВТ» А. Г. Гейн и др. 12 шт.;
- книга ученика, содержащая описания приемов работы и более двухсот задач и примеров 12 шт.;
- дискеты с программной поддержкой, примерами и задачами 2 шт.;
- справочно-методическое пособие для учителя, включающее в себя примерное тематическое и поурочное планирование согласно программ базового курса информатики, опубликованных в «ИНФО» №1 за 1993 г. 1 шт.

Пакет «Основы алгоритмизации»

Пакет позволяет быстро и качественно организовать изучение одноименного раздела в базовом курсе информатики.

Материал в книге учителя разбит по темам, для которых сформулированы: цель изучения, перечень необходимых знаний, умений и навыков, форма организации занятий, оснащение и, при необходимости, домашнее задание.

В книге ученика описаны учебные компьютерные среды. В этой статье мы кратко познакомимся с некоторыми из них.

Учебная компьютерная среда (УКС) «Кенгуренок-чертежник» облегчит учителю организацию изучения базовых алгоритмических конструкций и проверку знаний, умений и навыков учащихся по этой теме.

Симпатичный кенгуренок Ру живет в прямоугольнике, который мы будем называть полем (рис. 2). Ру может перемещаться по полю, но не как захочет, а только одинаковыми шагами в вертикальном и горизонтальном направлениях.



В непосредственном режиме кенгуренок сразу выполняет команду. В программном — сначала формируется набор команд (алгоритм), который затем исполняется.

Если непонятно, как работает тот или иной алгоритм, вы можете перейти в режим отладки, в котором перед выполнением каждой команды на экране появляется мальчик Вилли. Он укажет на ошибку или передаст команду Ру.



Вилли поможет вам проверить знания. В режиме прокрутки он будет задавать ученикам вопросы о порядке выполнения алгоритма: выполнено условие или нет, какая команда будет выполняться следующей...

Вилли сообщает, правильно ли дан ответ, и поправит при ошибке, сформирует сообщение-статистику об ошибках ученика (рис. 1).

Конструкция	Вопрос	Ошибка
ЕСЛИ		
условие	0	0
установка	0	0
ИНАЧЕ		
установка	0	0
КОНЕЦ ВЕТВЛЕНИЯ		
установка	0	0
ПОКА		
условие	0	0
установка	0	0
КОНЕЦ ЦИКЛА		
установка	0	0
ВЫЗОВ ПРОЦ.		
установка	5	3
КОНЕЦ ПРОЦ.		
установка	4	1

Рис. 1. Статистика ошибок

В режиме решения задачи (рис. 2) необходимо, во-первых, придумать алгоритм, эквивалентный заложенному в задаче, во-вторых, реализовать этот алгоритм на языке команд исполнителя. «Жизни» — « » покажут, сколько раз можно получить подсказку или выполнить сравнение. При вызове подсказки УКС показывает движение кенгуренка под управлением алгоритма,

K1прыжок
K2шаг
K3поворот
K4если
K5иначе
K6тогда
K7сделай
K8проектируй
K9конец...

прыжок
шаг
сделай зq
поворот
сделай Zs
сделай зq
поворот
сделай Zs
сделай gt
сделай Zs
сделай зq
поворот
сделай Zs
шаг
сделай gt
сделай Zs
сделай Zs
шаг
сделай gt
сделай Zs
сделай зq
поворот

Рис. 2. Вид экрана УКС «Кенгуренок-чертежник» в режиме «Решение задачи»

заложенного в задаче. В режиме сравнения одновременно показывается движение Ру под управлением написанного учеником алгоритма и под управлением алгоритма, заложенного в задаче.

Все задачи помещены в книгу ученика с указанием имен файлов. Управлять всеми компьютерными средами можно с помощью меню, расположенного в верхней части экрана (рис. 2).



Учебная компьютерная среда «Пылесосик Роби» — это хороший помощник в изучении темы «Работа с данными». Местом приложения сил пылесосика является прямоугольный стеллаж, состоящий из отдельных отсеков (рис. 3).

Каждый отсек может быть пуст или содержит один предмет. Карман (стек) пылесосика достаточно велик, что позволяет держать в нем одновременно несколько предметов, но он

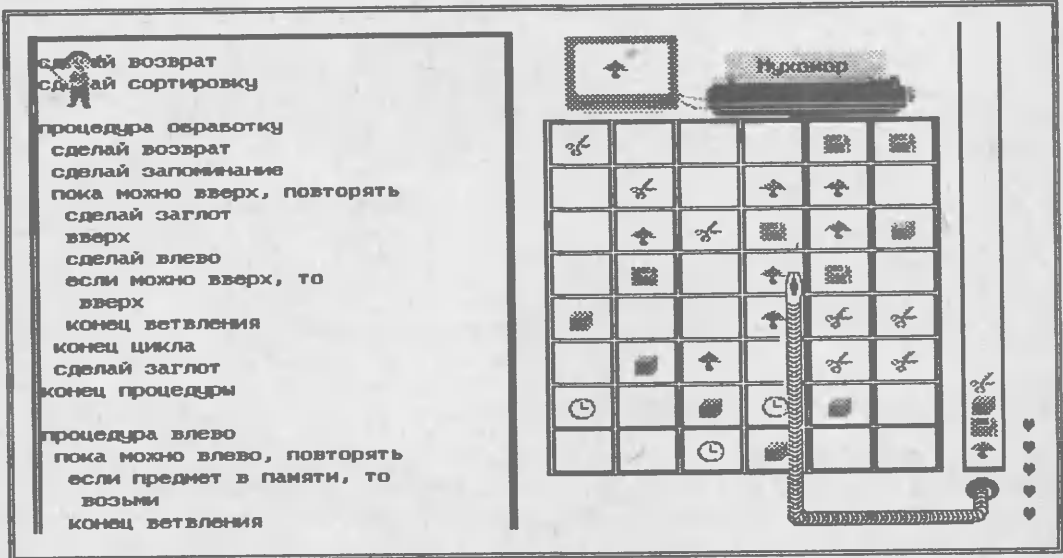


Рис. 3. Вид экрана УКС «Пылесосик Роби» в режиме отладки при решении задачи

устроен так, что предметы извлекаются из него в порядке, обратном их поступлению, т. е. сначала извлекается последний предмет, за ним предпоследний...

Голова Роби может перемещаться в любом из четырех направлений, переключать предмет из отсека в карман, из кармана в отсек.

Кроме того, Роби может запомнить предмет, находящийся в отсеке, расположенном под его головой, что позволяет ему осуществлять поиск однородных предметов и их сортировку в произвольном порядке.

Список проверяемых условий у Роби, по сравнению с Ру, значительно шире: пылесосик может проверить, пуст ли карман или отсек, на который указывает его голова, а также сравнить, совпадает ли предмет в текущем отсеке с тем, что хранится в его памяти. Как и кенгуренок, пылесосик может осуществлять ветвления и циклы, исполнять вспомогательные алгоритмы. Эти возможности данной УКС позволяют организовать с ее использованием изучение основных приемов работы с данными, таблицами и массивами, в частности перестановки и сортировки.

У Роби также есть друг. Это девочка, которую зовут Милли. Она делает для Роби то же, что для Ру делал Вилли: передает команды, спрашивает, выполняется ли то или иное условие, проверяет правильность написания алгоритма... За одним-единственным исключением: для Роби не реализован режим прокрутки. Это и не нужно, так как ученики уже знают базовые алгоритмические конструкции, отработав правильное владение ими в работе с Ру.

Система команд исполнителя «Пылесосик Роби» в точности соответствует системе команд исполнителя «Робот-манипулятор» из главы 6 учебника, поэтому адаптация не представляет никаких трудностей, и все задачи из учебника могут быть предложены учащимся для выполнения в описываемой УКС.

Пакет «Информатика — отрасль производства»

Пакет содержит набор УКС, справочник, упражнения, задания для самостоятельной работы, примерное тематическое и поурочное планирование для проведения занятий по теме «Информатика — отрасль производства».

Этот пакет рассчитан на использование в базовом курсе информатики для формирования у учащихся представлений об основных функциональных возможностях текстового редактора (ТР), электронной таблицы (ЭТ), информационно-поисковой системы (ИПС) в процессе их первичного изучения.

УКС «Текстовый редактор» и «Электронная таблица» имеют такой же интерфейс, как и в описанных выше компьютерных средах. Включение режима трассировки вызывает для объяснения и показа мультипликационный персонаж, который, комментируя свои действия, помогает ученикам освоить операции поиска и отладки, работу с блоками и макрооперации в ТР, проверить правильность расчетов в ЭТ.

Статистические ячейки ЭТ и ИПС покажут минимальное, максимальное, среднее значения или сумму текущего столбца.

В книгах для ученика и учителя приведены задачи на расчет заработной платы, другие задания, дан подробный конспект выполнения лабораторной работы № 18 из учебника.

На основе изучения и применения этих примеров вы сами сможете подготовить задания ученикам.

УКС «Информационно-поисковая система»

Информационно-поисковая система нашего ПМК создана для управления базами данных табличного вида. Размеры и количество записей и полей создаваемых баз данных могут изменяться. Управление, как и во всех УКС нашего ПМК, производится нажатием так называемых «горячих» клавиш, высвеченных в меню. Также во всех УКС имеется резидентный справочник (Help), вызываемый в любой момент одновременным нажатием клавиш <УПР> и <Н>.

Рассмотрим работу с информационно-поисковой системой на примере решения следующей задачи: в базе данных «Зоопарк» выясним, какие из зверей имеют в кличке букву «а» и одновременно весят больше 500 кг (рис. 4).

Для этого нажимаем <К1> — «запрос», и в верхней части экрана появляется строка для формирования запроса, в которой уже стоит наименование первого поля записи базы данных. Нажимаем <К3> — «следующий атрибут» до тех пор, пока не высветится поле с именем «Кличка». Затем нажимаем клавишу АВ: в строке запроса появился знак «=». Снова нажимаем клавишу <ТАВ> — после знака равенства появляется кавычка — приглашение к вводу маски. Вводим : *а*, где звездочка заменяет любое количество символов.

Вид	Пол	Кличка	Возраст	Вес	Питание
Слон африканский	самец	Билли	48	3428	Овоши
Лев африканский	самка	Белый	5	348	Мясо
Медведь бурый	самка	Валера	9	350	Рыба
Крокодил	самка	Гена	58	350	Рыба
Заяц белаяк	самка		2	4.5	Трава
Коршун	самец	Зорро	6	12	Мясо
Черепаша	самка		148	2.3	Овоши
Олень европейский	самец	Лася	12	115	Сена
Олень американский	самка	Эндрюша	8	180	Сена
Волк серый	самец	Лютый	4	56	Мясо
			max 140	min 2.3	

Рис. 4. Информационно-поисковая система в режиме отладки при выполнении запроса

Новое нажатие <ТАВ> вызывает появление логического союза «и», обозначающего «пересечение» простых запросов, составляющих сложный запрос.

Таким же образом вводим вторую часть запроса:

Кличка = * а * и Вес 500

Если нажать <К1> — «результат», то ИПС выделит подходящие записи. Для наглядности обучения вы можете наблюдать процесс выполнения запроса к базе данных в режиме отладки (рис. 4). Мультипликационный персонаж выполнит с комментариями все действия и тем самым подробно объяснит работу ИПС в режиме выполнения запроса.

Вы можете удалить выделенные или невыделенные записи, тем самым осуществив фильтрацию базы данных.

Используя фильтрацию и операцию «добавление базы», вы имеете возможность организовать ручную сортировку записей, как это описано в книге учителя. Опыт преподавания показал, что ученик, хоть раз вручную выполнивший операцию сортировки, уже не будет испытывать трудностей при работе с системами, где эти операции автоматизированы, поскольку у него будет создан наглядный образ правильного действия.

Краткие итоги

Эксплуатация комплекса в массовой школе показала, что в результате использования ПМК «IBM на УКНЦ» в преподавании базового курса информатики в 6—9 и даже в 4—5 классах средней школы наблюдаются следующие положительные эффекты:

- начинающие учителя практически сразу могут включиться в работу на должном уровне, так как они избавлены от необходимости самостоятельно разрабатывать планирование курса, подбирать программное обеспечение, составлять большое количество задач, разрабатывать «с нуля» методику объяснения материала;
- преподаватели избавлены от необходимости готовить дидактические материалы;
- автоматизация проверки знаний учащимися базовых алгоритмических конструкций и правильности решения задач экономит время учителя и гарантирует объективность оценки.

Более полное описание ПМК «IBM на УКНЦ» и методики его применения для организации занятий по базовому курсу информатики будет помещено в первом номере «Альманаха УКНЦ», который выйдет в первом квартале 1994 года в издательстве «Информатика и образование».

На все вопросы, связанные с этой статьей, вам ответят по телефону: (095) 155-87-30.

По вопросам приобретения звоните по телефону: (095) 155-87-37.

В первый понедельник каждого месяца Центр Интерактивных Средств Обучения Министерства Образования РФ проводит БЕСПЛАТНЫЕ семинары по изучению ПМК «IBM на УКНЦ». Мы можем также провести семинары на местах с выездом нашего специалиста.

НПК «ИНТЕРГАММА»

предлагает ШИРОКИЙ ВЫБОР системных, прикладных, обучающих и игровых ПРОГРАММ для ПЭВМ «Агат-7/9» и классов «Агат», «Электроника МС0511» (УКНЦ) и классов УКНЦ.

КАТАЛОГИ ПРОГРАММ МОЖНО ЗАПРОСИТЬ ПО ПОЧТЕ.

Наш адрес: 123081, Москва, а/я «НПК ИНТЕРГАММА».

О. Е. Баранов,
Фирма F-BIT

МОДЕМ ДЛЯ УКНЦ

Информатика сегодняшнего дня немыслима без средств электронных коммуникаций. С их помощью возможна не только своевременная транспортировка «горячей» информации или проведение электронных конференций, но и обучение, а также дистанционный контроль его результатов. При этом нет никакой разницы в том, каким обучающим средством прививаются и контролируются знания. Главное — чтобы каналы и «язык», на котором происходит обмен сообщениями о достигнутых результатах, совпадали.

Одним из наиболее распространенных каналов является телефонная сеть, а язык общения представлен в виде цифровых сигналов, имеющих допустимый для телефонной сети спектр частот. Аппарат, создающий возможность такого обмена информацией, принято называть модемом. Модем может представлять собой как отдельную законченную конструкцию, имеющую интерфейс с ПЭВМ и телефонной сетью, так и блок, устанавливаемый в ПЭВМ и дополняющий состав ее внешних устройств. В последнем случае модем называют внутренним. Такое решение позволяет получить минимальную стоимость за счет интеграции общих для всех электронных устройств аппаратных средств: источника питания, средств управления, индикации и т.д.

К настоящему моменту разработано множество модемов, доступных для подключения к самым распространенным ПЭВМ: IBM PC, Apple, Sinclair, Atari, PDP, VAX и многим другим. Тем не менее для пользователей ПЭВМ УКНЦ эти модемы практически недоступны из-за особенностей аппаратного построения. Решить эту проблему и призван модем, предлагаемый для этих ПЭВМ фирмой F-BIT, занимающейся разработкой программного обеспечения для образования и инвестицией средств в разработку аппаратного обеспечения для системы образования.

Что же представляет собой модем для УКНЦ?

Внутренний модем — это аппаратная приставка, устанавливаемая в свободный слот ПЭВМ УКНЦ. Разработана также и специальная программа управления аппаратурой модема, загружаемая в ОЗУ периферийного процессора. Управляющая программа запускается в коммуникационную оболочку через регистры связи канала K1.

Отличительной особенностью модема является то, что он имеет совместимость на уровне команд управления с Hayes-модемами, составляющими 80% всех модемов мира. То есть, как и любой другой из этого класса модемов, он управляется с помощью стандартного набора AT-команд. Эти команды позволяют сделать общение с модемом не сложнее, чем с принтером, а кроме того, обеспечивают независимость коммуникационной оболочки от типа применяемого модема. Модем отличается относительно низкой стоимостью при высоких эксплуатационных и технических характеристиках. Основные из них представлены в таблице.

Для передачи данных, включая почтовые сообщения и файлы программ, необходима коммуникационная оболочка, входящая в пакет программного обеспечения модема. Эта коммуникационная оболочка представляет собой программу гибкой и развитой архитектуры, предназначенную для взаимодействия с человеком как с потребителем информации, с одной стороны, и модемом для ее приема или передачи, с другой. Пересылка конечных структур осуществляется с помощью резидентных протоколов передачи: X-modem, Y-modem. Возможно подключение и внешних протоколов, а также программная эмуляция стандартов сжатия и исправления данных по MNP и V.42.

Специальное средство коммуникационной оболочки — терминальная программа — дает возможность проводить сеансы связи в режимах, принятых для «электронных досок объявлений», известных как станции BBS сети FIDONET и др. Для того чтобы совпадали кодировки символов, принятые в различных компьютерах и используемые в разных сетях, терминальная программа имеет несколько встроенных в нее эмуляторов терминалов, наиболее часто применяемых в ЭВМ. Эти эмуляторы дают возможность управлять режимом вывода информации способом, принятым для компьютера, с которым установлено соединение в данный момент.

Таблица

Асинхронная скорость	1200, 600, 300 бит/с			
Диапазон асинхронной скорости	+2.3% — 2.5% режим расширенного диапазона из стандарта ССИТТ в символах асинхронного режима			
Асинхронный формат	10...15 бит, включая 1...6 стартовых, 8 информационных и 1 стоповый бит			
Модуляция	Частотная ССИТТ V.23 при 1200 и 600 Бод, ССИТТ V.21 и BELL 103			
Частота несущей "1"	Режим	Передатчик	Приемник	
V.23 (1200 бит/с)	Вызывающий	1300 Гц	1300 Гц	
	Отвечающий	1300 Гц	1300 Гц	
V.23 (600 бит/с)	Вызывающий	1300 Гц	1300 Гц	
	Отвечающий	1300 Гц	1300 Гц	
V.21	Вызывающий	980 Гц	1650 Гц	
	Отвечающий	1650 Гц	1980 Гц	
BELL 103	Вызывающий	1270 Гц	2225 Гц	
	Отвечающий	2225 Гц	1270 Гц	
Частота несущей "0"	Режим	Передатчик	Приемник	
V.23 (1200 бит/с)	Вызывающий	2100 Гц	2100 Гц	
	Отвечающий	2100 Гц	2100 Гц	
V.23 (600 бит/с)	Вызывающий	1700 Гц	1700 Гц	
	Отвечающий	1700 Гц	1700 Гц	
V.21	Вызывающий	1180 Гц	1850 Гц	
	Отвечающий	1850 Гц	1180 Гц	
BELL 103	Вызывающий	1070 Гц	2025 Гц	
	Отвечающий	2025 Гц	1070 Гц	
Чувствительность приемника	Порог включения — 45 дБ			
	Порог выключения — 48 дБ			
Выравнивание уровня	Определением, при установке соединения методом среднего			
Доступная диагностика	Локальная аналоговая петля			
	Удаленная цифровая петля			
Генератор образов самодиагностики	Переменяющиеся единицы и нули с детектором ошибок. Число, указывающее количество ошибок, выводится			
Совместимость	Стандарт	Скорость	Режим	Метод
	ССИТТ V.23	1200 бит/с	Полудуплекс	Асинхронный
	СИТТ V.23	600 бит/с	Полудуплекс	Асинхронный
	ССИТТ V.21	300 бит/с	Полудуплекс	Асинхронный
	BELL 103	300 бит/с	Полудуплекс	Асинхронный
Режим работы	Непрерывный, в течение 96 часов + 2-часовой перерыв			
Конструктивное исполнение	Блок в стандарте ВУ УКНЦ, корпус — пластик			

Если этого недостаточно, можно использовать внешний эмулятор или не использовать вообще никакого, если соединенные ПЭВМ однотипны. Связь возможна с любыми удаленными ЭВМ, оснащенными совместимыми модемами.

Для передачи почтовых сообщений в автономном режиме существует еще один специальный режим коммуникационной программы. Система самостоятельно ответит на вызов или дозвонится куда-либо в заданное время, сохранит принятые почтовые сообщения и зафиксирует протокол своей работы в файле. Таким образом, модем может выполнять роль электронного почтового ящика.

Для того чтобы передача информации занимала минимальное время, в состав коммуникационной оболочки входит программа сжатия данных перед передачей и распаковки после приема. Эта же программа-архиватор может быть использована для сохранения данных (группы из текстовых или произвольных файлов) в более компактном виде.

Любой из режимов поддерживается «записной книжкой» — специальной базой, содержащей все данные о том, с кем необходимо связаться. С помощью программы-конвертора есть возможность пополнять эту базу информацией из распространяемых по сетям связи бюллетеней, которые содержат такую информацию, но в другом формате.

Пакет программ снабжен печатной документацией, рассчитанной на пользователей, не имеющих специальной подготовки. Таким образом, все, что потребуется для работы, — это подключение модема к компьютеру и телефонной сети, а также соблюдение некоторых простых правил при работе с коммуникационной программой.

Установка модема может открыть самые широкие перспективы. С его помощью станет возможным, например, проведение диспутов и олимпиад по любому из предметов. Проведение этих мероприятий возможно как со школой, находящейся на противоположном конце улицы, так и с колледжем на другой стороне планеты. И все это, не выходя из класса в течение обычных занятий! Можно смело утверждать, что модем и коммуникационная оболочка — это единственные средства, обеспечивающие выход в мир компьютерных коммуникаций для ПЭВМ УКНЦ, а также самое современное средство общения.

<p>КОМПЬЮТЕРНАЯ ПОДДЕРЖКА (НАЧАЛЬНАЯ ШКОЛА)</p> <ul style="list-style-type: none"> • МОЗАИКА • МАТРИС • CONTROL CAD • СЕТЕВОЕ ЛОГО • НАТУРАЛЬНЫЕ ЧИСЛА • WORD CONSTRUCTOR 	<p>ПРОГРАММНЫЕ СРЕДСТВА ПО ИНФОРМАТИКЕ</p> <ul style="list-style-type: none"> • РОБОТЛАНДИЯ • СРЕДА МПТ-ЛОГО • УЧЕБНАЯ БАЗА ДАННЫХ • ЭЛЕКТРОННЫЕ ТАБЛИЦЫ • ПАКЕТ МАШИНОПИСЬ • PAF-COMMANDER • FAST DRAW • АССИСТЕНТ 	<p>КОМПЬЮТЕРНАЯ ПОДДЕРЖКА ПРЕДМЕТОВ СРЕДНЕЙ ШКОЛЫ</p> <ul style="list-style-type: none"> • МАТЕМАТИКА • ГРАФИКИ ФУНКЦИЙ • МЕХАНИКА • CONTROL CAD
<p>ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТЧИКА</p> <ul style="list-style-type: none"> • ПАСКАЛЬ • СИ • ФОРТ • ГРАФИЧЕСКИЕ БИБЛИОТЕКИ VOLKOV'S TOOLS • ПАКЕТ UKGRAPH • ОТЛАДЧИК PULT 	<p>УКНЦ</p>	<p>Игровые пакеты: Игрушка 1-6 Эмулятор БК-0010</p>
<p>ТЕХНИЧЕСКОЕ ОСНАЩЕНИЕ КЛАССА</p> <ul style="list-style-type: none"> • ВИНЧЕСТЕР • ЭЛЕКТРОННЫЙ ДИСК • ТЕЛЕФОННЫЙ СЕРВЕР • МОДЕМ • ПРИНТЕР • ГРАФОПОСТРОИТЕЛЬ (ПЛОТТЕР) • МЫШЬ 		<p>ДЕЛОВЫЕ ПРИЛОЖЕНИЯ</p> <ul style="list-style-type: none"> • МИНИ-КАНЦЕЛЯРИЯ • МИНИ-БУХГАЛТЕРИЯ • СКЛАД • КОНВЕРТОР ИВМ-УКНЦ • АРХИВАТОР
<p>БЭИТ</p>	<p>(095) 535-2222 534-4831</p>	

НАМ ПИШУТ

Ю. М. Горвиц,

Научный директор Ассоциации «Компьютер и детство»

МЕЖДУНАРОДНЫЙ ДЕТСКИЙ КОМПЬЮТЕРНЫЙ ЛАГЕРЬ-ШКОЛА



Институт программных систем Российской академии наук и Русский факультет Всемирного университета регулярно организуют для одаренных и творчески активных детей летнюю компьютерную школу в Международном детском компьютерном лагере-школе (МДКЛШ).

Лагерь-школа расположен на берегу живописного Плещеева озера, в сосновом лесу, недалеко от старинного русского города Переславля-Залесского — в 150 километрах от Москвы. Этот лагерь известен во многих уголках России, в других государствах СНГ и за рубежом. Каждый год сюда приезжают ребята из США, Германии, Франции и других стран. Лагерь является ассоциированным членом ЮНЕСКО.

Летний учебный цикл обычно состоит из четырех-пяти смен, каждая из которых длится две или три недели.

В каждую смену работает несколько «кафедр», объединяющих учителей, их ассистентов и детей, которые совместно участвуют в проектах по выбранной тематике. При этом каждый учащийся сам выбирает тему занятий, в которой наилучшим образом раскрываются его творческие способности.

Учебные программы, возраст учащихся и уровень подготовки детей в разных сменах неодинаковы. В содержание летних учебных программ традиционно входят компьютерные науки, информатика, естественные и гуманитарные науки, литература и искусство, иностранные языки и другие дисциплины. Во многом содержание программ определяют преподаватели, в качестве которых в лагере работают ведущие ученые и специалисты научно-исследовательских институтов, академических и университетских центров, имеющие большой опыт научной и педагогической работы.

Техническую базу лагеря составляют в основном IBM-совместимые компьютеры и компьютеры «Ямаха». Программное обеспечение зависит от выбранной тематики кафедры и обычно предоставляется ее руководителем.

Используются традиционные универсальные текстовые, графические, музыкальные редакторы, настольные издательские системы, средства коммуникации, системы программирования C, Pascal, Lisp, Basic, система ЛОГО, а также оригинальные учебные, развивающие программы, способствующие разностороннему образованию детей.

Первая смена ориентирована на детей младшего школьного возраста (6—9 лет), не имеющих, за редким исключением, никакой компьютерной подготовки. Традиционно тема этой смены: знакомство с компьютером, первые познания в информатике, элементы программирования (в первую очередь ЛОГО и Бейсик), а также разнообразные обучающие и развивающие компьютерные игры.

Чем же занимались три недели 150 детей из Москвы и Пущино, Ярославля и Минска, Смоленска и Углича, Могилева и Переславль-Залесского?

Первая же вечерняя «планерка» с представлением руководителей кафедр и обсуждением предстоящей жизни показала, что собрались люди творческие, интересные, увлеченные, не побоюсь по прошествии времени сказать — талантливые. Ближайшие же несколько дней показали, что у этого состава преподавателей есть еще кое-что (поначалу неуловимое, а к концу смены ставшее явным) очень важное: полное взаимопонимание, какая-то общая благожелательная атмосфера и желание всем вместе сделать так, чтобы ребятам в лагере было интересно и весело.

На кафедре «Волшебная фея», которой руководила Ольга Куриленко из киевского НПО «Горсистемотехника», самые маленькие девочки и мальчики (6—7 лет) играли в развивающие компьютерные игры канадской компании DIL, рисовали картинки по компьютерным сюжетам, при этом еще и пели, и танцевали, и читали стихи. На лицах малышей всегда были улыбки, им явно нравилось все, что впервые происходит с ними.

Для детей постарше (8—10 лет) работала кафедра «Компьютер и детство». Ею руководил автор этой заметки Юрий Горвиц. Ежедневно ровно в 10.30 после завтрака стайка из 12—15 малышей уже с нетерпением поджидала меня у комнаты, где стояли компьютеры с программами ассоциации «Компьютер и детство». «Мои» дети с огромным удовольствием строили из чисел дома, спасали цыплят от хитрой лисы, ломали голову над тем, как же перехитрить сразу трех волков и провести поросят к водопою, чертили замысловатые узоры и рисовали фантастические картины, исследовали вулканы и землетрясения, ходили по компьютерной карте и еще много-много чего делали, не подозревая, что при этом развиваются. И уж конечно, не забудут они, как мастерски сражались в пространственные крестики-нолики: программа «КУБ-ИГРА» была одной из самых популярных.

Ведущей в лагере несомненно была кафедра «Роботландия» под руководством Ю. А. Первина, на которой с опытными специалистами А. Дувановым и И. Вертеевым занимались ребята постарше. Программы «Роботландии» давно и хорошо известны, о них много писали в ИНФО.

Необычной по теме были кафедры «Стрэнджерс» и «Мэджик ленд». У них и названия звучат не по-русски. Просто на этих кафедрах ребята с помощью увлекательных оригинальных программ под руководством замечательных преподавателей Надежды Прохоровой, Ольги Сенкевич и Надежды Первиной изучали английский и французский языки. Версии программ «Мэджик ленд» совершенствовались прямо по ходу их апробации на «кроliках» — детях из лагеря-школы.

Название кафедр отражало тематику занятий. Не исключение и кафедра «ЛОГО-мастер». Сотрудник Института информационных технологий в образовании Кирилл Александров сумел за короткий срок научить больше дюжины детей основным и многим дополнительным возможностям языка ЛОГО. Немало изобретательности проявили ученики «Лого-мастера», чтобы суметь самостоятельно создать несколько небольших мультипликатов в среде «LogoWriter», причем с довольно непростой логикой, цветных и сопровождавшихся музыкой.

К. Александров со своими помощниками руководил также пресс-центром, в котором дважды в неделю выпускалась компьютерная газета с последними новостями, а также издавалась различная иная оперативная полиграфическая продукция (чаще всего шуточная).

«Некомпьютерные» кафедры посещались детьми лагеря с не меньшим интересом (и слава богу, ведь наш лозунг — «Компьютер не цель, а средство!»). Охотно шли ребята и к Александру Ларионову, который учил их ремеслу резьбы по дереву и плетения корзин из ивы, и к Надежде Горюшкиной — замечательному музыкальному работнику, — которая за три недели выучивала играть на гитаре и которая помогла организовать и провести шесть концертов, и к Вячеславу Кравцову — мастеру керамики, чтобы вылепить из глины себе и друзьям сувениры, и к Лидии Векшиной — необыкновенной художнице, чтобы постигнуть тонкости изобразительного искусства. А какой детский лагерь без спорта! Будущий

преподаватель французского языка и замечательный гитарист Алексей Зозин вместе с будущим историком Мариной Баяновой (оба студенты Ярославского педагогического института) оказались еще и мастерами по организации спортивных соревнований.

Любимец всех детей, руководитель кафедры экологии биолог Максим Клепиков организовывал походы в лес, где дети знакомились с удивительными явлениями природы, узнавали о растениях и животных средней полосы России.

Занятия на компьютерных и других кафедрах обычно проводились в первой половине дня. А вечерами в любую погоду все с удовольствием посещали дискотеки, фильмы и концерты, которые готовили преподаватели вместе с детьми, и которые конечно же были на темы компьютерной жизни в лагере.

Работа каждой кафедры завершалась изготовлением и представлением на общей конференции проектов и изделий, сделанных детьми под руководством преподавателей, и награждением отличившихся детей призами, которые обычно устанавливают спонсоры лагеря.

Обращаюсь ко всем, кто увлекается компьютерами и программированием, кто любит заниматься с детьми чем угодно на базе информационных технологий, кто в каждом ребенке видит не только алгоритм, процессор с памятью, устройство ввода-вывода, но и собственно РЕБЕНКА, — милости просим!

Для желающих провести июньскую смену 1994 года в МДКЛШ сообщаем контактные телефоны:

В Москве: (095) 931-44-55

Горвиц Юрий Михайлович

В Переславль-Залесском: (08535) 7-05-10

Федоров Виктор Михайлович,

Карлик Михаил Исаакович

СМАРКЕТ

Еженедельная газета для пользователей, бизнесменов и программистов. Старейшая компьютерная газета на русском языке (выходит с 1990 г.).

*Каждую неделю
самая оперативная информация о важнейших событиях
в компьютерном мире России и СНГ,*

- информация о новых программных продуктах и технических средствах,
- обзорные материалы по проблемам компьютерной техники и ее применения,
- аналитические материалы о состоянии российского компьютерного рынка
- и многое другое... интересное и полезное.

Наш подписной индекс:

32015 для индивидуальных подписчиков

50138 для предприятий и организаций

ВОЗМОЖНА ПОДПИСКА ЧЕРЕЗ РЕДАКЦИЮ

Наш адрес:

127562, Москва, ул. Каргопольская, 17

Адрес для писем:

127562, Москва, а/я 120

Телефон:

(095) 903-21-47

Факс:

(095) 903-02-66

E-mail:

root@smarket.msk.su (via Relcom)

root@soft.zgrad.glas.apc.org (via GlasNet)

Д. М. Златопольский

КРОССВОРД

Предлагаем вниманию наших читателей кроссворд, который можно использовать как на уроке информатики, так и во внеклассной работе, с целью повышения интереса учащихся к предмету, расширения их кругозора. Ответы на этот кроссворд будут опубликованы в № 1 нашего журнала за 1994 г.



ВОПРОСЫ К КРОССВОРДУ

По горизонтали:

3. Язык программирования. 5. Точка подключения внешних устройств к внутренней шине микропроцессора. 6. Специальная область памяти персональных компьютеров. 7. Марка персональных компьютеров. 8. Знак арифметической операции. 10. Форма взаимодействия пользователя с компьютером.

По вертикали:

1. Единица измерения информации. 2. Цифра десятичной системы исчисления. 3. Фирма-разработчик систем «Турбо» для языков программирования. 4. Место на диске, в котором хранятся имена файлов и их другие характеристики. 9. Название клавиши. 11. Характеристика файла или переменной величины.

Если тема «Кроссворд» вам кажется интересной, присылайте свои разработки в редакцию.

ВСЕМ РУКОВОДИТЕЛЯМ ШКОЛ, ПТУ, ТЕХНИКУМОВ, УЧИТЕЛЯМ И МЕТОДИСТАМ,

использующим вычислительную технику в учебном процессе

Не спешите расходовать огромные средства на IBM-совместимые учебные классы. Вы можете сэкономить и найти Вашим деньгам лучшее применение!

Ваши терминальные классы КУВТ-86, УКНЦ, «Корвет» еще могут хорошо послужить и обеспечить Ваш учебный процесс на уровне современных требований. Для этого достаточно укомплектовать эти классы комплексными программно-методическими пакетами.

!!! СРЕДНЯЯ СТОИМОСТЬ ОДНОГО ПАКЕТА — 30 000 РУБЛЕЙ !!!

**НАУЧНО-ТЕХНИЧЕСКОЕ ПРЕДПРИЯТИЕ «НОВАЯ АЛЬТЕРНАТИВА» ПРЕДЛАГАЕТ ВАМ
ПРИБРОСТИ ПО САМЫМ УМЕРЕННЫМ ЦЕНАМ СЛЕДУЮЩИЕ
ПРОГРАММНО-МЕТОДИЧЕСКИЕ ПАКЕТЫ:**

Для классов КУВТ-86 (любых типов):

1. Пакет АЛЬТЕРНАТИВА-2 (КУВТ). Назначение — изучение информатики. Состав — 7 полностью записанных дисков. Включает: новый сетевой монитор, обеспечивающий все виды пересылок, ТУРБО-ПРОЛОГ, новейшие версии текстовых, музыкальных, графических редакторов, новую систему управления базами данных, клавиатурные тренажеры, электронные таблицы, исполнители, пакет новых игровых программ, подробную техническую и методическую документацию.
2. Пакеты учебных программ РОБОТЛАНДИЯ и ЛОГО, предназначенные для изучения основ информатики с младшими школьниками. Являются полными аналогами пакетов, хорошо себя зарекомендовавших на классах ЯМАХА.
3. Система программирования ТУРБО-ПАСКАЛЬ, подобна используемой на IBM PC.

Для класса УКНЦ (любых типов):

Пакет АЛЬТЕРНАТИВА-2 (УКНЦ). Назначение — изучение информатики в старших классах. Состав — 5 полностью записанных дисков. Включает: файловый монитор типа NORTON, сетевой монитор, обеспечивающий все типы пересылок по сети, текстовые, музыкальные, графические редакторы, систему управления базами данных, систему ТУРБО-ПАСКАЛЬ с графикой, транслирующую в коды, ТУРБО-ПРОЛОГ, электронные таблицы с графическим выводом, пакет игровых программ, подробную техническую и методическую документацию.

Для классов «Корвет» (любых типов):

Пакет АЛЬТЕРНАТИВА-2 («Корвет»). Назначение — обеспечение изучения информатики в старших классах. Состав — 5 полностью записанных дисков. Включает: сетевой монитор, обеспечивающий быструю пересылку по сети операционной системы CP/M и все виды работы с сетью; текстовый, музыкальный, графический редакторы; систему управления базами данных, систему ПАСКАЛЬ с графикой, транслирующую в коды; электронные таблицы, систему ТУРБО-ПРОЛОГ, пакет игровых программ, подробную техническую и методическую документацию.
Мы ОБЪЯВЛЯЕМ АМНИСТИЮ всем НЕЗАКОННЫМ и НЕЗАРЕГИСТРИРОВАННЫМ пользователям наших пакетов и просим их всех обратиться за получением новых версий. Мы ОБЪЯВЛЯЕМ о своей готовности передать на льготных условиях наши инструментальные и программные средства учителям и методистам, способным создать с их помощью для нас новые программные и методические разработки.

**Наша ставка — на высокое качество и большое число заказчиков,
а не на высокие цены.**

Наши пакеты — это то, что Вам доступно и так необходимо сегодня!

Не теряйте драгоценного времени. ВЫШЛИТЕ НАМ ЗАЯВКУ СЕГОДНЯ ЖЕ!

Наш адрес: 636057, г.Барнаул, а/я 2513. **НОВАЯ АЛЬТЕРНАТИВА.**

Директор Гриценко А.Н.



НА ВСЕ НАШИ ИЗДАНИЯ МОЖНО ПОДПИСАТЬСЯ ПО БЕЗНАЛИЧНОМУ РАСЧЕТУ

У вас в руках последний выпуск журнала «Информатика и образование» прошлого года — года, растянувшегося для редакции на все 500 дней. Заключительные два номера были для нас убыточными. Но коллектив редакции постарался сделать все, чтобы выполнить обязательства перед вами — нашими подписчиками. Удалось обойтись без сдвигания номеров и сокращения объема. Вот только исчез второй цвет внутри журнала, и выходит он с опозданием... Хотелось бы только отметить, что фактически опаздывала текущая цифра номера журнала, материалы же в них печатались самые «свежие».

Однако первый номер 1994 года уже в типографии, оригинал-макет второго готов. А далее наш «экспресс» будет следовать строго по графику.

Коллектив редакции очень благодарен подписчикам, которые сохранили верность нашему журналу и действительно помогают нам выживать в это нелегкое время.

В 1994 году мы также гарантируем выпуск всех шести номеров. Что же будет в них напечатано?

Читатели увидят в журнале не только традиционные рубрики. Мы, конечно же, будем поддерживать пользователей отечественной техники, которой еще очень много в школах России. Учителя, работающие с БК, УКНЦ, «Корветом», «Агатом», а также «Ямахой», найдут в нашем журнале нужную им информацию.

Естественно, остаются и методические материалы, и педагогический опыт, и задачи...

Но часть журнала будет отдана современной технике, технологии, программным продуктам. Читатели, наверное, оценили новую рубрику «Профессиональная ПЭВМ», в которой публикуются статьи по IBM-совместимым компьютерам и их применению в системе образования, о мощных компьютерах, рабочих станциях, новых пакетах прикладных программ, новой периферии и электронике. Будут также открыты рубрики «Мультимедиа в образовании», «Телекоммуникации» и др.

Кроме того, планируется публиковать материалы по истории вычислительной техники, обзоры по «софту» и «харду», статьи российских и зарубежных ученых. Но в большой степени содержание и политику журнала формируете вы, наши читатели. Пишите, каким вы хотите видеть журнал, какие материалы необходимы вам для работы.

Теперь о нашей гордости — о приложениях. Несмотря на все тяготы жизни, редакция решилась не только поставить на подписку, но и издать два приложения к журналу «Информатика и образование». Это — «Персональный компьютер БК-0010 — БК-0011М» и «Персональный компьютер УКНЦ».

Первый (и единственный) номер журнала «Персональный компьютер БК-0010 — БК-0011М» за 1993 год уже выпущен и разослан подписчикам. Первый номер 1994 года выйдет в апреле, второй — в июне.

Первый номер журнала «Персональный компьютер УКНЦ» выйдет в мае этого года.

Во втором полугодии журнал «Персональный компьютер БК-0010 — БК-0011М» будет выходить один раз в два месяца, а «Персональный компьютер УКНЦ» — один раз в три месяца.

ЕЩЕ РАЗ ОБРАЩАЕМ ВНИМАНИЕ ЧИТАТЕЛЕЙ, ЧТО НА ВСЕ НАШИ ИЗДАНИЯ МОЖНО ПОДПИСАТЬСЯ ПО БЕЗНАЛИЧНОМУ РАСЧЕТУ.

Сведения, необходимые для подписки на наши издания на второе полугодие 1994 года:

Название журнала	Подписной индекс	Периодичность	Стоимость подписки по каталогу «Роспечати»
«Информатика и образование»	70423 (для индивидуальных подписчиков)	1 раз в 2 месяца	18 000 руб.
	73176 (для предприятий и организаций)		36 000 руб.
Библиотека журнала «Информатика и образование»			
«Персональный компьютер БК-0010 — БК-0011М»	73177 (для индивидуальных подписчиков)	1 раз в 2 месяца	12 000 руб.
	73092 (для предприятий и организаций)		21 000 руб.
«Персональный компьютер УКНЦ»	73179 (для индивидуальных подписчиков)	1 раз в 3 месяца	10 000 руб.
	73093 (для предприятий и организаций)		20 000 руб.



WRITER

PAINTER

PHOTOMASTER

ANIMATOR

IMAGER

DESIGNER

Multivision позволяет совершенно неподготовленным компьютерным пользователям создавать презентации быстро и просто, не изучая ни одного языка программирования. Создание интерактивных и саможивущих презентаций, включающих мультфильмы, звук, текст и графику - дело нескольких минут.

С помощью Multivision Вы можете создать интерактивные презентации, демонстрационные ролики, обучающие программы и многое другое.

Writer - это специализированный редактор для подготовки красочных экранных текстов. Позволяет импортировать тексты в формате Word Perfect 5.1, MS Word 5.0, Word for Windows 2.0, Write for Windows.

Painter - это удобный инструмент для рисования. Изображения в форматах PCX, PIC, PIC (Lotus), CLP (Quattro), BMP и GIF могут быть легко импортированы из Ваших любимых графических пакетов.

Библиотека, включающая более 1500 высококачественных изображений, касающихся сферы бизнеса, архитектуры, компьютеров и других сторон человеческой жизни, даст Вам необходимые дополнительные возможности.

PhotoMaster служит для обработки сканированных изображений в формате TIFF.

Animator - это простой инструмент, предназначенный для создания анимации. Вы только задаете ключевые кадры, их преобразования, траектории движения и количество промежуточных кадров, а все остальное система сделает автоматически.

Imager собирает картинки и мультфильмы в единые библиотеки и позволяет легко позиционировать на экране. Данный модуль позволяет проигрывать несколько мультфильмов одновременно.

Designer - это основной модуль, связывающий сцены в единую демонстрацию, представляющую единый выполняемый файл для DOS или Windows. Designer запишет Вашу демонстрацию в сжатом виде и снабдит программой установки на другой компьютер.



Основные характеристики:
Простой "point-and-click" интерфейс.
Множество шрифтов, символов и звуков.
Графический режим: 640x480x16 цветов.
Графические форматы: TIFF, PCX, PIC, PIC (Lotus), CLP (Quattro), BMP и GIF.
Поддержка текста в форматах MS Word 5.0, Word Perfect 5.1, Word for Windows, Write for Windows.

Полное руководство пользователя.
Учебное руководство и набор файлов для легкого и быстрого обучения.
Примеры.

Требования к аппаратуре:
IBM PC 286, PS/2 и старше или совместимые.
640 Kb RAM. VGA адаптер.
Мышь фирмы Microsoft или совместимое устройство.

MS-DOS версия 3.0 или старше.
7 Mb жесткого диска для установки Multivision.
10 Mb жесткого диска для установки библиотек картинок и символов.

IST

International Software Trading AB

Box 2093 350 02 VAXJO SWEDEN

Telephone: +46-470-171 00

AIST

Advanced Instructional Software Trading AB

103050 K-50 a/n 80 AICT AB Москва Россия Телефон 095-229 67 06

Работа с MultiVision требует немного, но дает намного больше.

MULTIVISION



MultiVision -
это удобный
Конструктор
Презентаций

45-132

MultiVision - это пакет программ
для деловых людей, преподавателей
и специалистов в любой области.

MULTIVISION



AIST Advanced Instructional
Software Trading AB